JSC-18820

Data Processing System Systems Brief

INTRODUCTION	1
GPC-AP-101B	2
GPC-AP-101S	3
MULTIFUNCTION CRT DISPLAY SYSTEM	4
MASS MEMORY UNIT	5
MASTER TIMING UNIT	6
MULTIPLEXER/ DEMULTIPLEXER	7
B/U FLIGHT CONTLR AND BFS/PASS INTERFACES	8
IPL-AP-101B	9
IPL-AP-101S	10
PASS SYSTEM SERVICES OVERVIEW	11
PASS INPUT/OUTPUT	12
SYSTEM MANAGEMENT	13
BACKUP SYSTEMS SOFTWARE	14
BFS INPUT/OUTPUT	15
BFS SYSTEMS MANAGEMENT	16
APPENDIX	A
APPENDIX	B

Mission Operations Directorate
System Division

Basic, Rev B, PCN-5
June 15, 1992

NASA
National Aeronautics and Space Administration

Lyndon B. Johnson Space Center
Houston, Texas

MISSION OPERATIONS DIRECTORATE

DATA PROCESSING SYSTEM
SYSTEM BRIEFS

BASIC, REV B, PCN-5

June 15, 1992

PREPARED BY

Robert Hudson
Data Processing System Section

APPROVED BY

Mark Erminger, Head
Data Processing System Section

Harold Clancy, Chief
Communication and Data Systems Branch

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

DATA PROCESSING SYSTEM
SYSTEM BRIEFS

BASIC, REV B

PREFACE

This document has been prepared by the Data Processing System Section, Communication and Data Systems Branch, Systems Division, NASA/Johnson Space Center, Houston, Texas. Any subsequent changes to this document will be printed as required.

Comments or questions concerning this document should be directed to DF22/ Robert Hudson, Data Processing System Section, (713) 244-5391.

MISSION OPERATIONS DIRECTORATE

DATA PROCESSING SYSTEM
SYSTEM BRIEFS

BASIC, REV B, PCN-4

June 20, 1991

PREPARED BY

John D'Agostino
Data Processing System Section

APPROVED BY

FOR Mark Erminger, Head
Data Processing System Section

Harold Clancy, Chief
Communication and Data Systems Branch

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

DATA PROCESSING SYSTEM
SYSTEM BRIEFS

BASIC, REV B

PREFACE

This document has been prepared by the Data Processing System Section, Communication and Data Systems Branch, Systems Division, NASA/Johnson Space Center, Houston, Texas. Any subsequent changes to this document will be printed as required.

Comments or questions concerning this document should be directed to DF22/ John D'Agostino, Data Processing System Section, 713/483-0545.

DATA PROCESSING SYSTEM
SYSTEM BRIEFS

CONTENTS

Section		Page
1	INTRODUCTION	1-1
1.1	SCOPE	1-1
1.2	PURPOSE	1-1
1.3	GENERAL LRU CONFIGURATION	1-2
1.4	REFERENCES	1-2

HARDWARE

2	GPC - AP-101B	2-1
3	GPC - AP-101S	TBS
4	MULTIFUNCTION CRT DISPLAY SYSTEM	4-1
5	MASS MEMORY UNIT	5-1
6	MASTER TIMING UNIT	6-1
7	MULTIPLEXER/DEMULTIPLEXER	7-1

BACKUP FLIGHT CONTROLLER AND BFS/PASS INTERFACES

8	BACKUP FLIGHT CONTROLLER AND BFS/PASS INTERFACES	8-1

SOFTWARE

9	IPL - AP-101B	9-1
10	IPL - AP-101S	TBS

PRIMARY AVIONICS SYSTEMS SOFTWARE

	PASS SYSTEM SERVICES OVERVIEW	11-1
11.1	FCOS	11.1-1
11.1.1	Time Management	11.1.1-1
11.1.2	Error Log Reset	11.1.2-1
11.1.3	HISAM Dump Software	11.1.3-1
11.1.4	Common and Redundant Set Synchronization	11.1.4-1
11.1.5	G3 Archive	11.1.5-1

Section		Page
11.2	USER INTERFACE	11.2-1
11.2.1	Uplink Software	11.2.1-1
11.2.2	Downlink Software	11.2.2-1
11.2.3	Software Dump	11.2.3-1
11.3	SYSTEM CONTROL	11.3-1
11.3.1	ICC Buffer and ICC Sumword Logic	11.3.1-1
11.3.2	OPS Transitions	11.3.2-1
11.3.3	GPC Initialization	11.3.3-1
12	PASS INPUT/OUTPUT	12-1
13	SYSTEMS MANAGEMENT	13-1

BACKUP FLIGHT SOFTWARE

14	BACKUP SYSTEMS SOFTWARE	14-1
15	BFS INPUT/OUTPUT	TBS
16	BFS SYSTEMS MANAGEMENT	16-1

Appendix

A	SELF TEST PROGRAM (STP) ERROR MESSAGES	A-1
B	BACKUP SYSTEM LOADER (BSL) ERROR MESSSAGES	B-1
C	INDEX	C-1

DATA PROCESSING SYSTEM
SYSTEM BRIEF

CONTENTS

Section		Page
1	INTRODUCTION	1.1-1
1.1	SCOPE	1.1-1
1.2	PURPOSE	1.2-1
1.3	GENERAL LRU CONFIGURATION	1.3-1

HARDWARE

2	GPC - AP-101B	2-1
3	GPC - AP-101S	TBS
4	MULTIFUNCTION CRT DISPLAY SYSTEM	4-1
5	MASS MEMORY UNIT	5-1
6	MASTER TIMING UNIT	6-1
7	MULTIPLEXER/DEMULTIPLEXER	7-1

BACKUP FLIGHT CONTROLLER AND BFS/PASS INTERFACES

8	BACKUP FLIGHT CONTROLLER AND BFS/PASS INTERFACES	8-1

SOFTWARE

9	IPL - AP-101B	9-1
10	IPL - AP-101S	TBS

PRIMARY AVIONICS SYSTEMS SOFTWARE

11	PASS SYSTEM SERVICES OVERVIEW	11-1
11.1	FCOS	11.1-1
11.1.1	Time Management	11.1.1-1
11.1.2	Error Log Reset	TBS
11.1.3	HISAM Dump	11.1.3-1
11.1.4	Common and Redundant Set Synchronication	11.1.4-1

Section		Page
11.2	USER INTERFACE	11.2-1
11.2.1	Uplink Software	TBS
11.2.2	Downlink Software	11.2.2-1
11.2.3	Software Dump	11.2.3-1
11.3	SYSTEM CONTROL	11.3-1
11.3.1	ICC Buffer and ICC Sumword Logic	11.3.1-1
11.3.2	OPS Transitions	TBS
11.3.3	GPC Initialization	11.3.3-1
12	PASS INPUT/OUTPUT	12-1
13	SYSTEMS MANAGEMENT	13-1

BACKUP FLIGHT SOFTWARE

14	BACKUP SYSTEMS SOFTWARE	14-1
15	BFS INPUT/OUTPUT	TBS
16	BFS SYSTEMS MANAGEMENT	16-1

1 INTRODUCTION

1 INTRODUCTION

SECTION 1
INTRODUCTION

CONTENTS

Section		Page
1.1	SCOPE	1-1
1.2	PURPOSE	1-1
1.3	GENERAL LRU CONFIGURATION	1-2
1.4	REFERENCES	1-2

SECTION 1
INTRODUCTION

1.1 SCOPE

This new edition of the Data Processing System - System Briefs, Basic Revision B, is totally reorganized. This new edition includes not only hardware, but also software system briefs. Moreover, where applicable, failure analysis has been added to the hardware briefs to document corporate knowledge on failure modes. The data in this book should not be considered as source data. Please refer to the sources when they are supplied. Eventually there will be appendices containing a bibliography and a key word index.

This new edition is divided into one minor and two major sections. First, the INTRODUCTION section relays to the reader the scope and purpose of this book and also presents an overview schematic of the data processing system. Second, the HARDWARE portion of the book contains briefs on the six major hardware elements of the DPS. Both the AP-101B and the AP-101S computers are covered in this section. The new SOFTWARE section of the book is divided into three sub-sections including initial program load descriptions for both GPCs, primary avionics software system, and backup flight software. The PASS and BFS sections are further subdivided to cover system services, input/output, and systems management software.

Please submit all corrections and comments for improvement to me so that periodic updates can improve the quality of this book. Please be patient while the new software section matures.

Data Processing System
System Briefs
Book Manager
John D'Agostino
Rockwell Shuttle Operations Company
Johnson Space Center
DF22
Houston, Texas 77058

1.2 PURPOSE

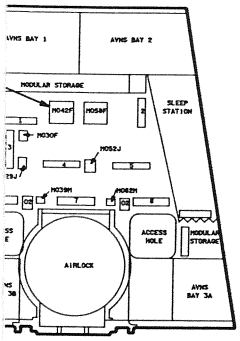
The purpose of this book is to condense a large amount of hardware vendor specifications, IBM and Rockwell software specifications, and DPS developed information into a book that is operationally functional. Hopefully, the new edition of software briefs to this document will improve its value to the operational engineer. Please refer to the vendor documents and drawings if there is a question concerning the validity of the contents.

1.3 GENERAL LRU CONFIGURATION

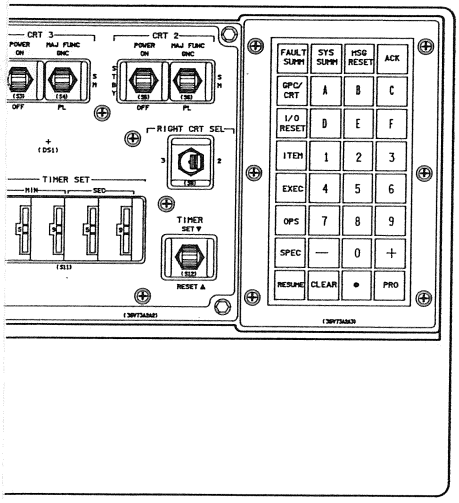
See figures 1-1, 1-2, and 1-3.

1.4 REFERENCES

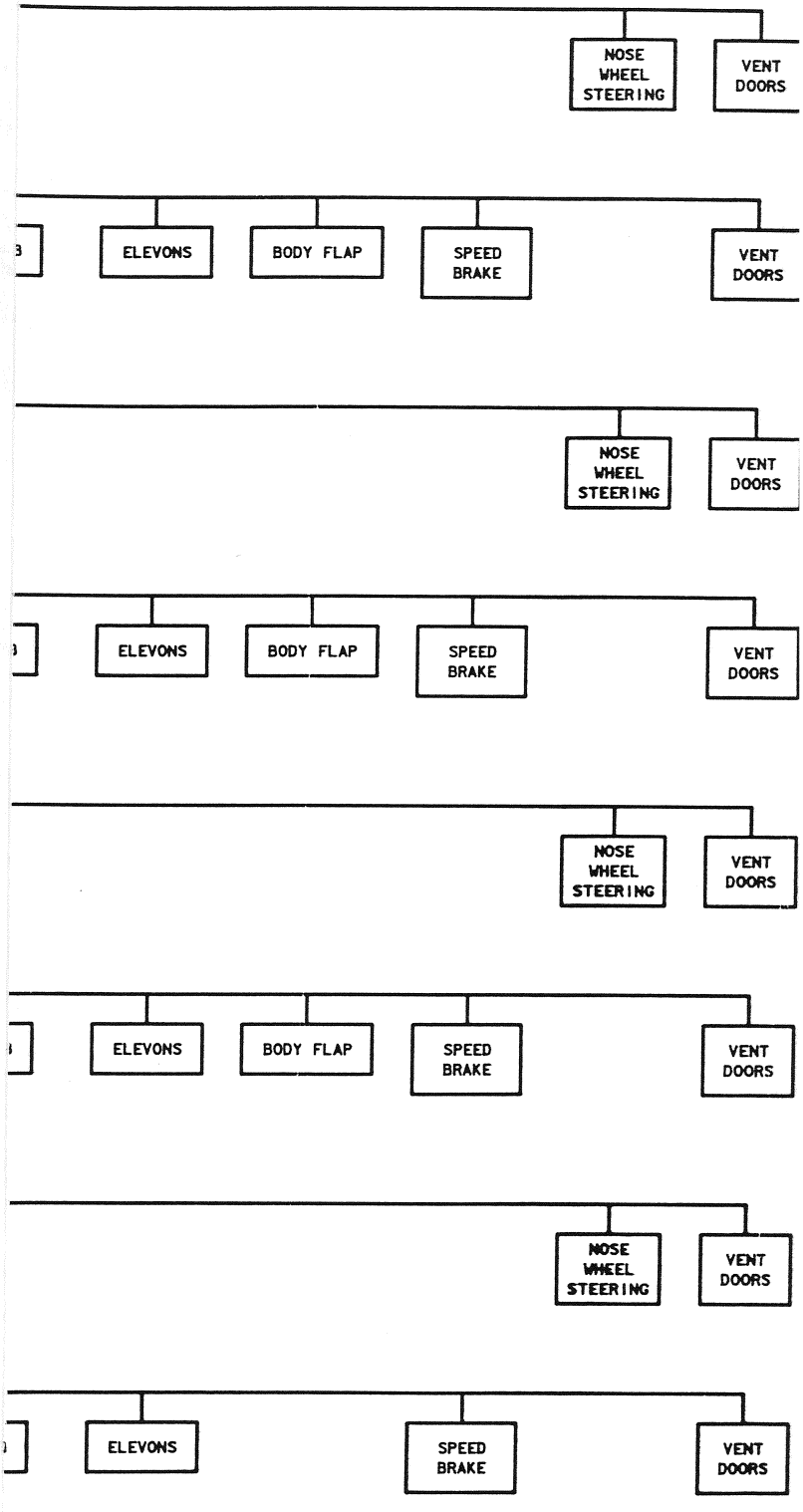
References for the sources of data in this book are annotated thrugh the use of superscripts. Superscript numbers will generally be found at the end of a section or subsection title, table titles, and figure titles. Superscripts refer to source documents listed in the REFERENCES subsection. This subsection is typically the last subsection in each section.



MID DECK CEILING
(LOOKING DOWN)



ıel 1, Control and Displays.



188200102.ART, 1

Figure 1-2.- Panel 2, flight-critical buses.

-3.- Panel 3, Non-Flight-Critical Buses.

SECTION 2
GPC-AP-101B

Section		Page
2.1	OVERVIEW	2-1
2.1.1	Component Locations	2-1
2.1.2	Crew Controls (Hardware Switches)	2-4
2.1.3	Power	2-4
2.1.4	Cooling	2-4
2.1.5	Connectors	2-7
2.2	CENTRAL PROCESSING UNIT	2-28
2.2.1	Overview	2-28
2.2.2	Main Memory	2-28
2.2.3	Direct I/O, Direct Memory Access (DMA) Channel	2-30
2.2.4	CPU Data Formats	2-31
2.2.5	CPU Microprogram	2-31
2.2.6	CPU General Purpose Registers (Local Store)	2-33
2.2.7	CPU Inner Data Flow	2-34
2.2.8	Interrupts	2-37
2.2.9	CPU Extended Addressing	2-50
2.2.10	CPU Real-Time Clock Timers	2-50
2.2.11	CPU Fault Detection	2-51
2.2.12	CPU Packaging Design	2-52
2.3	INPUT/OUTPUT PROCESSOR (IOP)	2-56
2.3.1	IOP Overview	2-56
2.3.2	IOP Channel Control	2-57
2.3.3	IOP Control Monitor	2-64
2.3.4	IOP Processor Elements	2-65
2.3.5	IOP MIA	2-69
2.3.6	IOP Redundancy Management Logic	2-69
2.4	GPC DISCRETE INPUT/OUTPUT	2-71
2.4.1	Discret Input/Output Circuits	2-71
2.4.2	GPC IOP Input Discrete Hardware Commonality	2-75
2.4.3	GPC Fail Vote Discretes/CAM Interface	2-77
2.4.4	PASS GPC Discrete Input RM	2-80
2.4.5	Failure of GPC Mode Discrete Inputs	2-81
2.4.5.1	GPC Mode Discrete Circuitry	2-81
2.4.5.2	HALT Discrete Failure	2-83
2.4.5.3	STBY Discrete Failure	2-83
2.4.5.4	RUN Discrete Failure	2-84
2.5	REFERENCES	2-85

SECTION 2
GPC-AP-101B

2.1 OVERVIEW

Each GPC is a modified IBM AP-101 CPU with a specially designed IOP which interfaces with the serial data bus network. These two line replaceable units (LRUs) of the GPC both contain portions of the main memory which are used by either LRU on a nondedicated basis. The CPU initiates all input/output actions through the execution of instructions to the IOP. These instructions and data words are transferred between the CPU and IOP on a bidirectional parallel word data bus.

The CPU is assigned the task of computation associated with the application programs. The CPU also provides clock synchronization, central control and handling of interrupts, and program control of IOP input/output requests. In fixed point, it is capable of 400,000 full word operations per second (four indexed full word adds to one indexed full word multiply).

An IOP is asociated with each CPU to provide control and monitoring of data bus traffic, and thereby relieve the CPU of these time consuming operations. All data transmissions between GPC's, communications on the serial data buses, and discretes into and out of the GPC are handled by the IOP. Each IOP bus interface can operate independently of the others within that IOP. Each IOP has the capability of interfacing with 24 serial data buses. The data buses that the IOP actually monitors or controls are determined by software assignments. IOP data processing programs, which are stored in common main memory, are independent of CPU instructions and have their own unique instruction set. The CPU controls the IOP by instructions issued over an input/output parallel data bus which connects the CPU and the IOP.

2.1.1 Component Locations

The GPCs are located in the pressurized portion of the Orbiter in avionics bays 1, 2, and 3A. Figures 2-1 and 2-2 show these AV bays and the positions of the GPC's and other equipment in them.

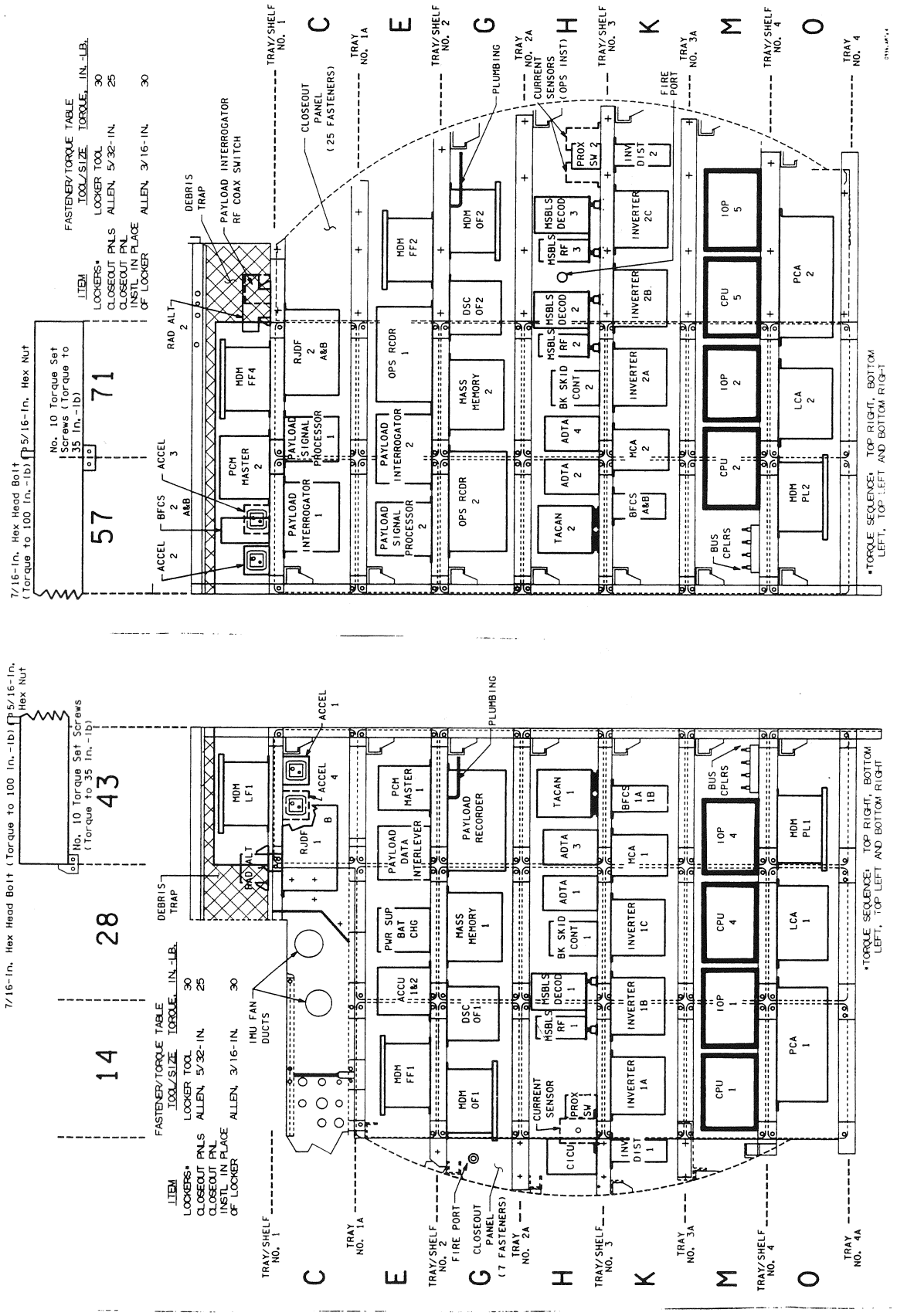


Figure 2-1.- AV bays 1 and 2.

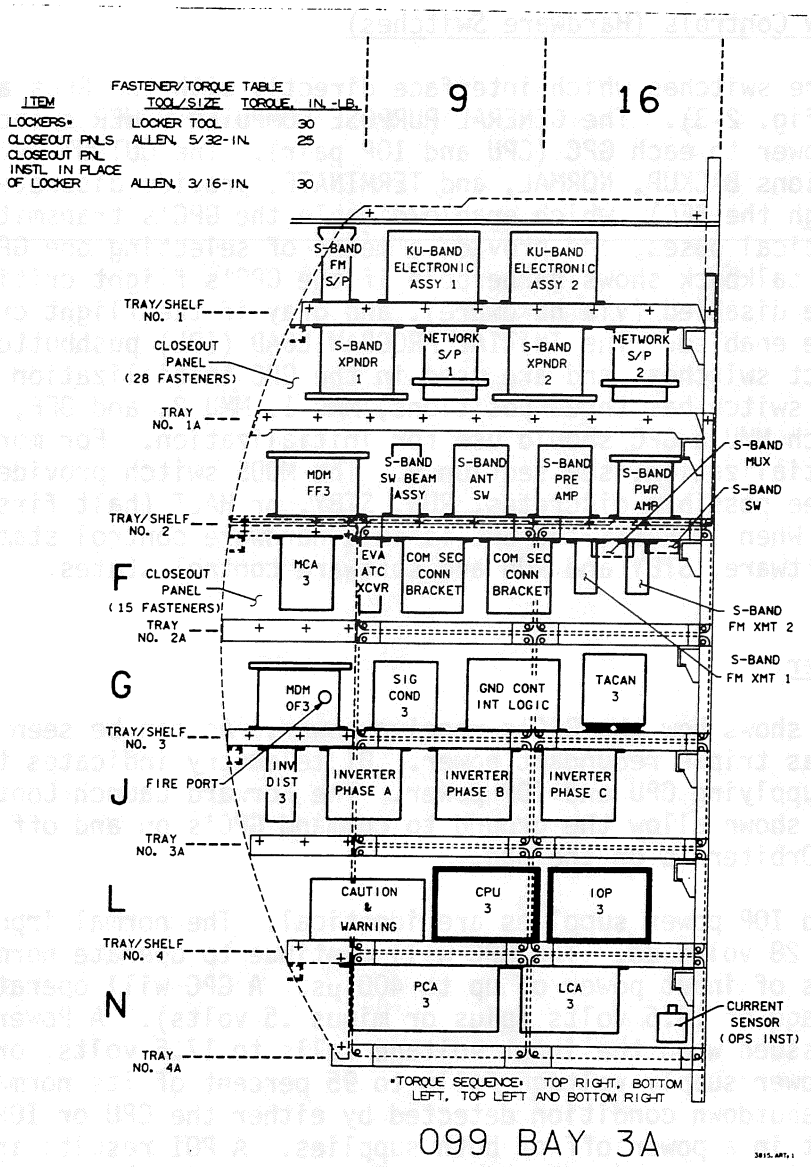


Figure 2-2.- AV bay 3A.

2.1.2 Crew Controls (Hardware Switches)

All hardware switches which interface directly with the GPCs are found on panel O6 (fig. 2-3). The GENERAL PURPOSE COMPUTER POWER switches 1 through 5 supply power to each GPC (CPU and IOP pair). The OUTPUT switches, which have positions BACKUP, NORMAL, and TERMINATE, provide discrete inputs to the GPC (through the BFC), which enable/disable the GPC's transmitters on the flight critical buses, and provide a means of selecting one GPC as the BFS. The OUTPUT talkback shows barberpole if the GPC's flight critical transmitters are disabled (via hardware), and gray if the flight critical transmitters are enabled. The INITIAL PROGRAM LOAD (IPL) pushbuttons are momentary contact switches, and are used in the GPC initialization process. The IPL SOURCE switch has three positions, MMU 1, MMU 2, and OFF, and is used to select which MMU a GPC should use for initialization. For more information on GPC initialization, see section 9. The MODE switch provides the GPC with one of three possible discretes, RUN, STBY, or HALT (halt first goes through the BFC). When in HALT, the GPC is in a hardware control state and cannot execute software, STBY and RUN are software control states.

2.1.3 Power

Figure 2-4 shows how the GPC's receive power. As can be seen in the figure, each GPC has triple redundant power. OI telemetry indicates the status of each RPC supplying CPU and IOP power. The Forward Launch Control Assemblies (FWD LCAs) shown allow the ground to command GPC's on and off through LF1 while the Orbiter is on the pad.

The CPU and IOP power supplies are identical. The normal input voltage for the GPC is 28 volts dc. The GPC will continue to operate normally for interruptions of input power of up to 400 μs. A GPC will operate at a steady state voltage of 18.5 volts (plus or minus .5 volts). A Power Off Interrupt (POI) is issued when the input voltage falls to 17.5 volts, or when any GPC internal power supply voltage falls to 95 percent of its normal operating value. A shutdown condition detected by either the CPU or IOP power supply will result in a power off of both supplies. A POI results in an orderly GPC shutdown wherein the CPU Program Status Word (PSW) and registers are stored in memory. The GPC response to powerup after a POI is software dependent. (A PASS GPC cannot be used if powered off in RUN or STBY. The BFS has a power-on restart capability and may be used even if it is powered off in a state other than HALT.)

2.1.4 Cooling

GPCs are cooled with air which is circulated through the AV bay by AV bay fans. There are two fans in each AV bay, and normally only one fan in each bay is powered.

Thermal tests were conducted to determine how long a GPC might be expected to run in a loss of cooling case. In the test, a GPC was allowed to run

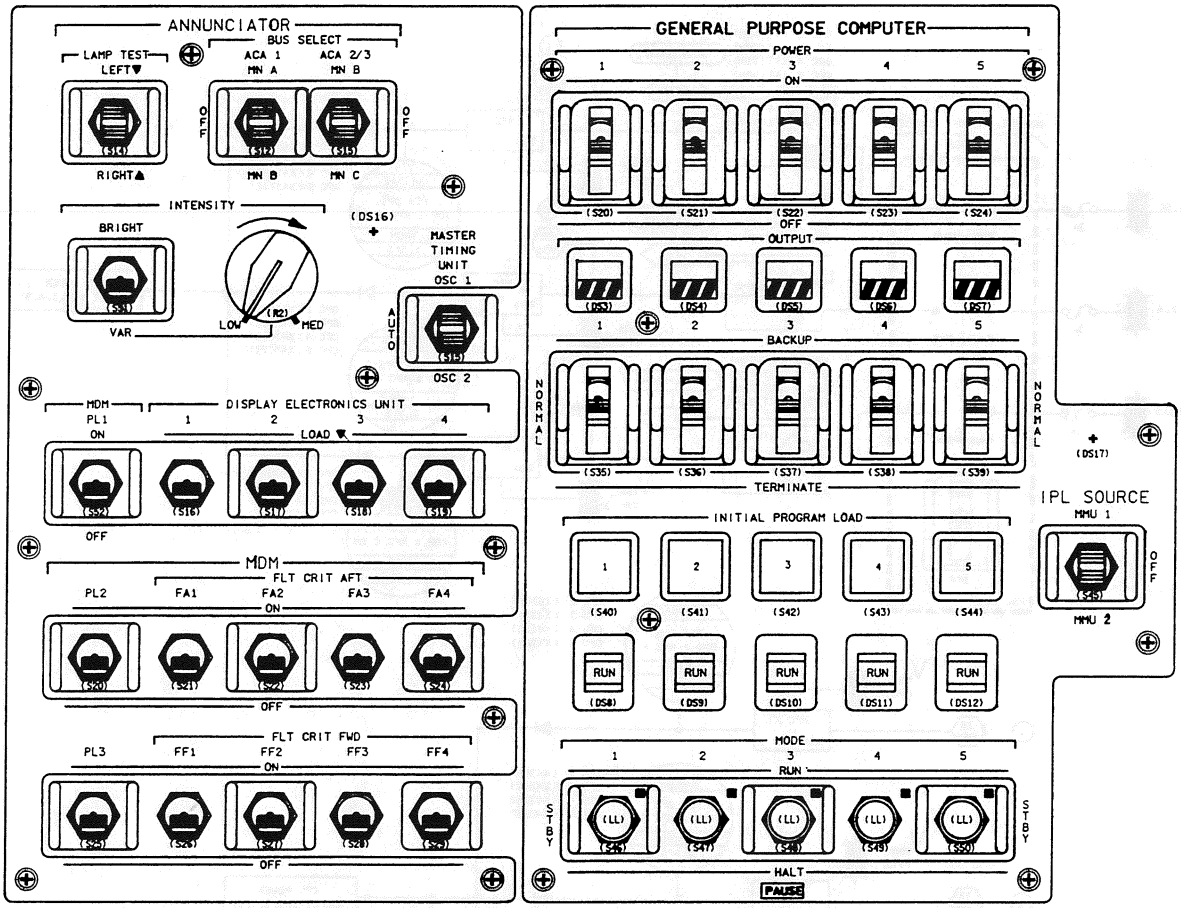


Figure 2-3.- GPC H/W switches (panel O6).

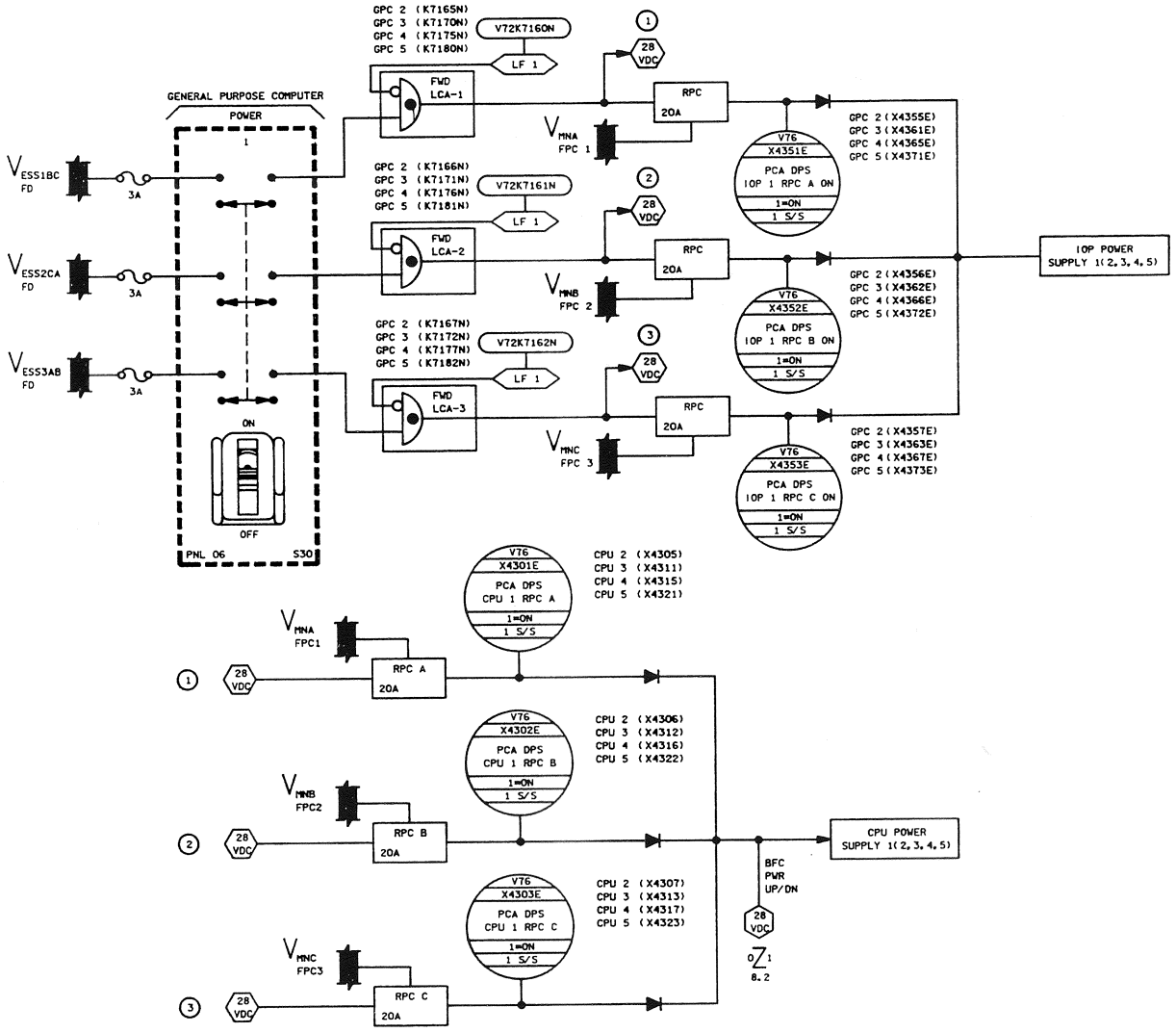


Figure 2-4.- GPC power.

until its hottest component reached the maximum specification temperature (125° C). A GPC should continue to operate up to a maximum component temperature of 150° C. Therefore, the data from the thermal tests were extrapolated to determine how long it should take to reach a maximum temperature of 150° C. The information from the tests is shown in table 2-I. The Flight Data File procedures for loss of cooling cases were built using the 150° data.

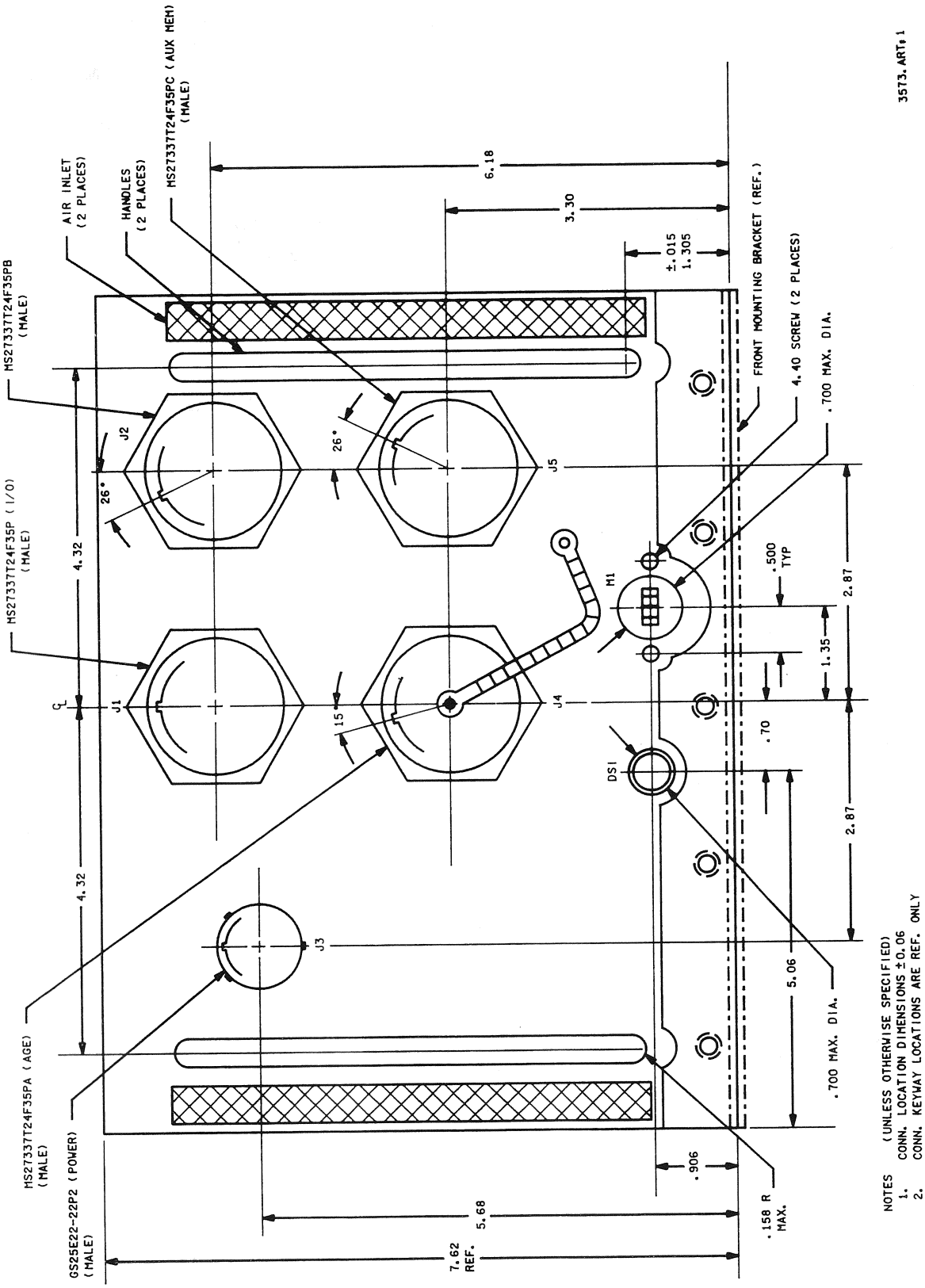
The term "cold soak" means to let the GPC cool off while being powered off until it reaches ambient temperature. A GPC which is turned off as soon as cooling is lost will be cold soaked in approximately 3.5 hours. A GPC which is allowed to continue operating after cooling is lost until its maximum component temperature is 125° C, will be cold soaked after being off for approximately 4.5 hours.

TABLE 2-I.- GPC LOSS OF COOLING OPERATION LIMITS

Initial conditions at time cooling lost	Operating time (minutes) to reach	
	125° C	150° C
GPC powered on and at normal operating temp, 1 atm pressure	26	60
GPC cold soaked, 1 atm pressure	56	90
GPC powered on and at normal operating temp, 0 atm pressure	20	50
GPC cold soaked, 0 atm pressure	45	75

2.1.5 Connectors

Figures 2-5 and 2-6 show the connector locations for the CPU and IOP. Figure 2-7 shows the pin assignments for a typical 128-pin connector used in the GPC. Figure 2-8 shows the pin assignments for the four-pin power connector used in the CPU and IOP. Table 2-II defines the signals for each of the connectors' pins.



NOTES (UNLESS OTHERWISE SPECIFIED)
1. CONN. LOCATION DIMENSIONS ±0.06
2. CONN. KEYWAY LOCATIONS ARE REF. ONLY

Figure 2-5.- CPU connector and bracket configuration.

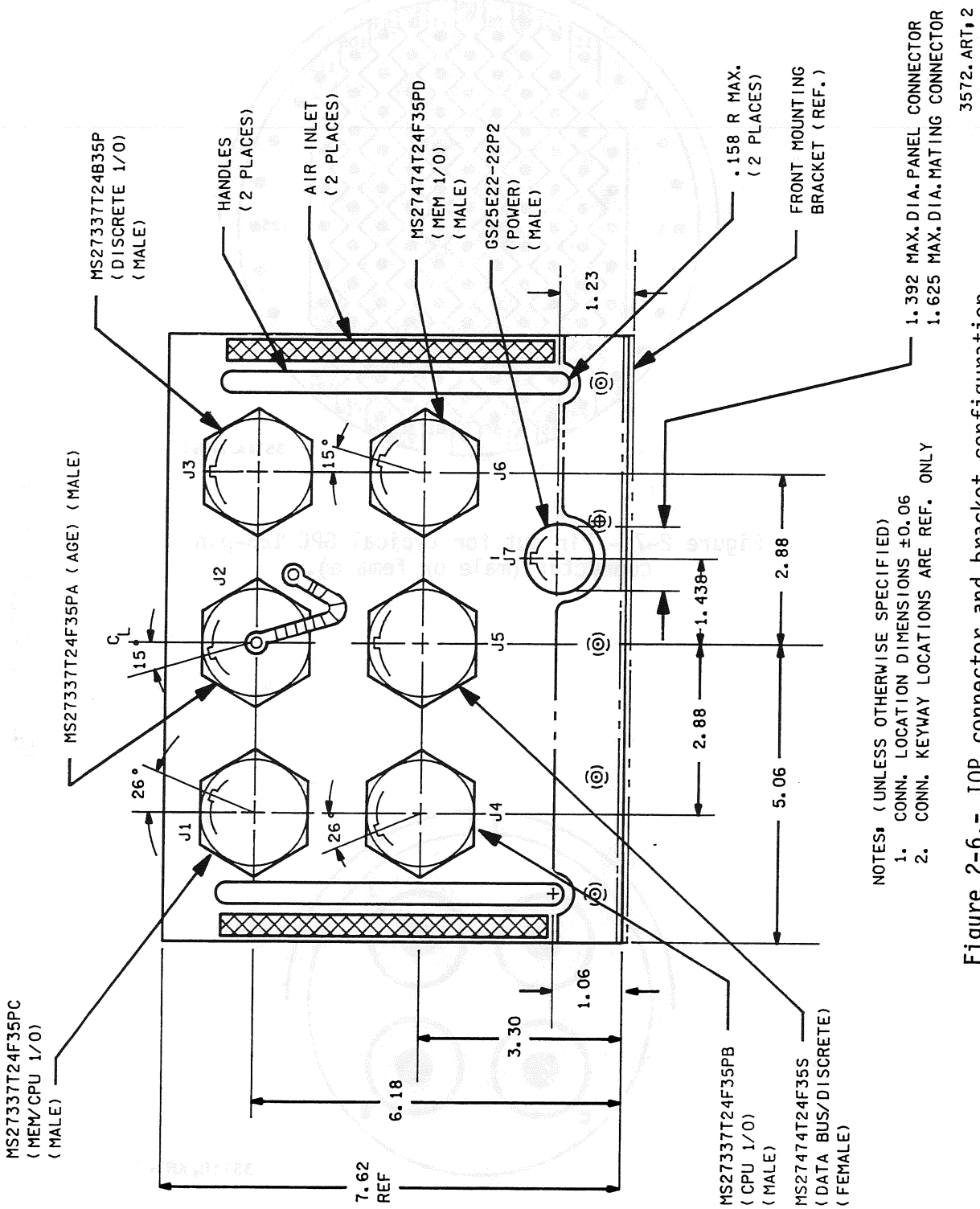


Figure 2-6.- IOP connector and bracket configuration.

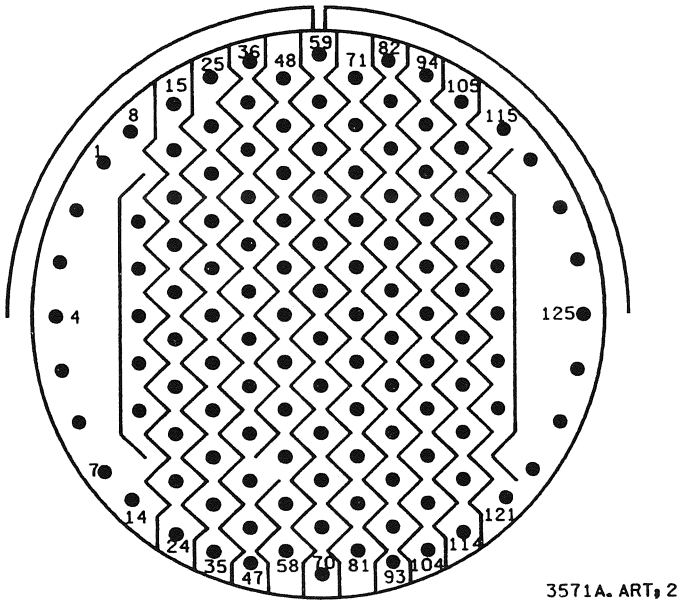


Figure 2-7.- Pin out for typical GPC 128-pin connector (male or female).

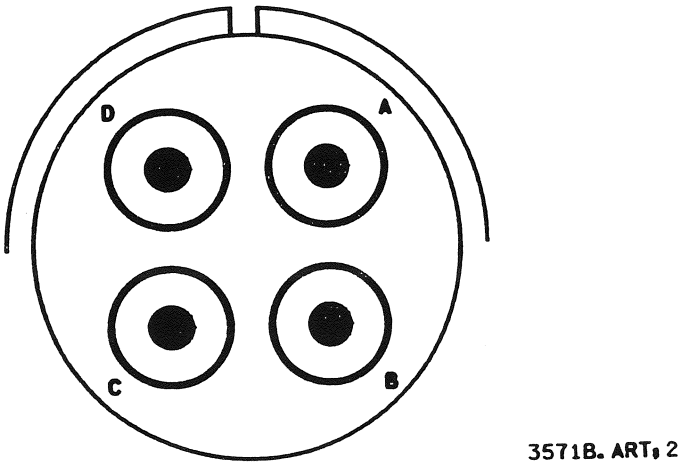


Figure 2-8.- Pin out for CPU IOP four-pin connector.

TABLE 2-II.- CONNECTOR/SIGNAL LIST

CPU connector J1 - connector part No. MS27337T24F35P IOP connector J4 - connector part No. MS27337T24FPB			
CPU J1 Pin No.	Signal name	Function	IOP J4 pin No.
37	EM00NT	+Data bus bit 00T	32
38	EM00NC	-Data bus bit 00C	21
20	EM01NT	+Data bus bit 01T	68
12	EM01NC	-Data bus bit 01C	80
23	EM02NT	+Data bus bit 02T	55
24	EM02NC	-Data bus bit 02C	44
22	EM03NT	+Data bus bit 03T	46
33	EN03NC	-Data bus bit 03C	57
36	EM04NT	+Data bus bit 04T	43
48	EM04NC	-Data bus bit 04C	31
52	EM05NT	+Data bus bit 05T	5
53	EM05NC	-Data bus bit 05C	12
100	EM06NT	+Data bus bit 06T	20
111	EM06NC	-Data bus bit 06C	19
104	EM07NT	+Data bus bit 07T	13
103	EM07NC	-Data bus bit 07C	6
92	EM08NT	+Data bus bit 08T	7
93	EM08NC	-Data bus bit 08C	22
40	EM09NT	+Data bus bit 09T	42
51	EM09NC	-Data bus bit 09C	30
101	EM10NT	+Data bus bit 10T	23
89	EM10NC	-Data bus bit 10C	14
69	EM11NT	+Data bus bit 11T	33
70	EM11NC	-Data bus bit 11C	34
43	EM12NT	+Data bus bit 12T	35
32	EM12NC	-Data bus bit 12C	24
47	EM13NT	+Data bus bit 13T	56
46	EM13NC	-Data bus bit 13C	45
125	EM14NT	+Data bus bit 14T	58
117	EM14NC	-Data bus bit 14C	47
108	EM15NT	+Data bus bit 15T	66
107	EM15NC	-Data bus bit 15C	54
4	EM16NT	+Data bus bit 16T	69
10	EM16NC	-Data bus bit 16C	70
118	EM17NT	+Data bus bit 17T	78
126	EM17NC	-Data bus bit 17C	77
9	EM18NT	+Data bus bit 18T	81
18	EM18NC	-Data bus bit 18C	93
121	EM19NT	+Data bus bit 19T	104
128	EM19NC	-Data bus bit 19C	92
119	EM20NT	+Data bus bit 20T	91
127	EM20NC	-Data bus bit 20C	102
73	EM21NT	+Data bus bit 21T	103
72	EM21NC	-Data bus bit 21C	114
8	EM22NT	+Data bus bit 22T	121
16	EM22NC	-Data bus bit 22C	113
114	EM23NT	+Data bus bit 23T	89

TABLE 2-II.- Continued

CPU J1 Pin No.	Signal name	Function	IOP J4 pin No.
113	EM23NC	-Data bus bit 23C	90
15	EM24NT	+Data bus bit 24T	128
25	EM24NC	-Data bus bit 24C	112
98	GP1NT	Gated priority DMA 01T	101
99	GP1NC	Gated priority DMA 01C	127
71	GP7NT	Gated priority DMA 07T	79
59	GP7NC	Gated priority DMA 07C	67
85	EM25NT	+Data bus bit 25T	88
96	EM25NC	-Data bus bit 25C	87
116	EM26NT	+Data bus bit 26T	126
124	EM26NC	-Data bus bit 26C	120
13	EM27NT	+Data bus bit 27T	110
6	EM27NC	-Data bus bit 27C	99
17	EM28NT	+Data bus bit 28T	111
1	EM28NC	-Data bus bit 28C	100
80	EM29NT	+Data bus bit 29T	109
81	EM29NC	-Data bus bit 29C	98
109	EM30NT	+Data bus bit 30T	125
110	EM30NC	+Data bus bit 30C	119
120	EM31NT	+Data bus bit 31T	86
112	EM32NC	-Data bus bit 31C	85
56	EMPLNT	+Data bus parity low T	118
68	EMPLNC	-Data bus parity low C	124
7	EMPHNT	+Data bus parity high T	116
14	EMPHNC	-Data bus parity high C	117
31	STPRNT	Set store protect control T	123
21	STPRNC	Set store protect control C	122
60	MPOR	Master power off reset	108
77	SYRNT	System reset T	107
88	SYRNC	System reset C	115
78	STP1NT	Stop control 01T	84
66	STP1NC	Stop control 01C	96
39	STP2NT	Stop control 02T	105
28	STP2NC	Stop control 02C	106
115	CONTNT	Continue T	95
106	CONTNC	Continue C	94
5	DTRONT	Data request I/O T	76
11	DTRONC	Data request I/O C	65
19	STRCNT	Store control T	83
29	STRCNC	Store control C	82
61	IPOINT	IOP power off INT T	71
50	IPOINC	IOP power off INT C	72
102	GLKONT	Gated lockout T	61
91	GLKONC	Gated lockout C	62
34	I1MSNT	IOP inhibit main store T	59
35	I1MSNC	IOP inhibit main store C	60
45	SPR2NT	Spare discrete 02T	49
44	SPR2NC	Spare discrete 02C	48
27	SPR1NT	Spare discrete 01T	37

TABLE 2-II.- Continued

CPU J1 Pin No.	Signal name	Function	IOP J4 pin No.
26	SPR1NC	Spare discrete 01C	36
30	GRDNT	Gate recvrs and drivers T	39
41	GRDNC	Gate recvrs and drivers C	51
67	1_{μ}CLNT	1 MHz clock T	25
55	1_{μ}CLNC	1 MHz clock C	15
90	DTACNT	Data acknowledge T	26
79	DTACNC	Data acknowledge C	27
58	RSIFNT	Reset interface functions T	16
57	RSIFNC	Reset interface functions C	8
42	GDV1NT	Gate device (address) T	40
54	GDV1NC	Gate device (address) C	52
64	TERMNT	Terminate T	1
65	TERMNC	Terminate C	17
3	MCHRNT	Machine reset T	9
2	MCHRNC	Machine reset C	2
95	DDMANT	Disable DMA T	3
84	DDMANC	Disable DMA C	10

J1 spares	T	C	J4 spares	T	C
1	63	62	1	29	41
2	75	74	2	11	4
3	76	87	3	18	28
4	83	82	4	38	50
5	97	86	5	53	64
6	94	105	6	74	73
7	123	122	7	75	63

Also pin 49 on J1 is a spare and pin 97 on J4 is a spare.

TABLE 2-II.- Continued

CPU connector J2 - connector part No. MS27337T24F35PB IOP connector J1 - connector part No. MS27337T24F35PC			
CPU J2 pin No.	Signal name	Function	IOP J1 pin No.
84	CB00NT	Address bus bit 00S (MSB)	82
83	CB00NC	Address bus bit 00C (MSB)	71
119	CB01NT	Address bus bit 01T	62
110	CB01NC	Address bus bit 01C	61
114	CB02NT	Address bus bit 02T	94
113	CB02NC	Address bus bit 02C	83
102	CB03NT	Address bus bit 03T	105
112	CB03NC	Address bus bit 03C	95
71	CB04NT	Address bus bit 04T	84
82	CB04NC	Address bus bit 04C	85
76	CB05NT	Address bus bit 05T	115
75	CB05NC	Address bus bit 05C	106
31	CB06NT	Address bus bit 06T	107
21	CB06NC	Address bus bit 06C	96
35	CB07NT	Address bus bit 07T	90
34	CB07NC	Address bus bit 07C	89
46	CB08NT	Address bus bit 08T	91
47	CB08NC	Address bus bit 08C	102
74	CB09NT	Address bus bit 09T	114
86	CB09NC	Address bus bit 09C	113
32	CB10NT	Address bus bit 10T	103
43	CB10NC	Address bus bit 10C	104
70	CB11NT	Address bus bit 11T	79
69	CB11NC	Address bus bit 11C	78
101	CB12NT	Address bus bit 12T	92
89	CB12NC	Address bus bit 12C	93
92	CB13NT	Address bus bit 13T	81
93	CB13NC	Address bus bit 13C	80
10	CB14NT	Address bus bit 14T	68
4	CB14NC	Address bus bit 14C	67
17	CB15NT	Address bus bit 15T	69
18	CB15NC	Address bus bit 15C	70
125	CBM1NT	Address bus bit M1T	56
117	CBM1NC	Address bus bit M1C	55
5	CBM2NT	Address bus bit M2T	58
11	CBM2NC	Address bus bit M2C	47
116	HWCNT	Halfword control, True	57
108	HWCNC	Halfword control, Complement	45
7	CBPNT	Address bus parity, True	46
14	CBPNC	Address bus parity, Complement	35
6	DPELNT	Output DMA parity error low T	59
12	DPELNC	Output DMA parity error low C	60
49	DPEHNT	Output DMA parity error high T	72
50	DPEHNC	Output DMA parity error high C	73

TABLE 2-II.- Continued

CPU J2 Pin No.	Signal name	Function	IOP J1 pin No.
105	AC02NT	AUX address bit 02T	97
94	AC02NC	AUX address bit 02C	86
30	AC03NT	AUX address bit 03T	66
29	AC03NC	AUX address bit 03C	54
59	AC04NT	AUX address bit 04T	29
48	AC04NC	AUX address bit 04C	18
27	AC05NT	AUX address bit 05T	43
39	AC05NC	AUX address bit 05C	42
3	AC06NT	AUX address bit 06T	41
9	AC06NC	AUX address bit 06C	40
120	AC07NT	AUX address bit 07T	53
127	AC07NC	AUX address bit 07C	64
107	AC08NT	AUX address bit 08T	20
122	AC08NC	AUX address bit 08C	31
20	AC09NT	AUX address bit 09T	65
19	AC09NC	AUX address bit 09C	77
13	AC10NT	AUX address bit 010T	87
22	AC10NC	AUX address bit 010C	75
68	AC11NT	AUX address bit 011T	112
79	AC11NC	AUX address bit 011C	121
121	AC12NT	AUX address bit 012T	111
128	AC12NC	AUX address bit 012C	101
111	AC13NT	AUX address bit 013T	120
100	AC13NC	AUX address bit 013C	119
72	AC14NT	AUX address bit 014T	108
60	AC14NC	AUX address bit 014C	98
54	AC15NC	AUX address bit 015T	122
42	AC15NC	AUX address bit 015C	123
55	ACM1NT	AUX address bit M1T	74
66	ACM1NC	AUX address bit M1C	63
85	ACM2NT	AUX address bit M2T	110
97	ACM2NC	AUX address bit M2C	100
8	ACCONT	Address bus bit COI-T	99
16	ACCONC	Address bus bit COI-C	109
126	ACPNT	AUX address bit parity T	88
118	ACPNC	AUX address bit parity C	76
109	XNT0NT	External interrupt 00T	24
98	XNT0NC	External interrupt 00C	34
61	XNT1NT	External interrupt 01T	44
73	XNT1NC	External interrupt 01C	33
45	XNT2NT	External interrupt 02T	23
33	XNT2NC	External interrupt 02C	14
103	XNT3NT	External interrupt 03T	7
104	XNT3NC	External interrupt 03C	22
91	XNT4NT	External interrupt 04T	21
90	XNT4NC	External interrupt 04C	32
96	DMA1NT	DMA request T	13
95	DMA1NC	DMA request C	6
99	MAC1NT	Microprogram break-in request T	5
87	MAC1NC	Microprogram break-in request C	12

TABLE 2-II.- Continued

CPU J2 Pin No.	Signal name	Function	IOP J1 pin No.
67	SP1NNT	SP input microbranch T	30
78	SP1NNC	SP input microbranch C	19
81	IMFNT	IOP fail (BITE) T	3
80	IMFNC	IOP fail (BITE) C	10
88	CPUFNT	CPU fail (BITE) T	51
77	CPUFNC	CPU fail (BITE) C	52
124	XDDSNT	IOP transmitter disable T	1
123	XDDSNC	IOP transmitter disable C	17
26	BDDSNT	BCE disable T	27
38	BDDSNC	BCE disable C	28
2	GCWNT	Gated control word T	16
1	GCWNC	Gated control word C	8

J2 spares	T	C	J1 spares	T	C
1	25	15	1	39	38
2	36	37	2	26	15
3	28	40	3	25	37
4	52	51	4	49	50
5	53	41	5	36	48
6	57	58	6*	11	4
7	63	62	7*	9	2
8*	56	44			
9*	64	65			

Also IOP J1 pins 117, 118, 125, 126 - not connected to tape cable.

*Reserved for EMU.

TABLE 2-II.- Continued

CPU connector J5 - connector part No. MS27337T24F35PC IOP connector J6 - connector part No. MS27474T24F35PD			
CPU J5 pin No.	Signal name	Function	IOP J6 pin No.
48	AM08NT	AUX data bus bit 08T	48
36	AM08NC	AUX data bus bit 08C	36
37	ESELNT	Select memory T	37
25	ESELNC	Select memory C	25
26	ASELNT	AUX select T	26
15	ASELNC	AUX select C	15
17	AM10NT	AUX data bus bit 10T	17
1	AM10NC	AUX data bus bit 10C	1
9	AM11NT	AUX data bus bit 11T	9
2	AM11NC	AUX data bus bit 11C	2
10	AM16NT	AUX data bus bit 16T	10
3	AM16NC	AUX data bus bit 16C	3
22	AM17NT	AUX data bus bit 17T	22
20	AM17NC	AUX data bus bit 17C	20
13	AM29NT	AUX data bus bit 29T	13
12	AM29NC	AUX data bus bit 29C	12
5	AM30NT	AUX data bus bit 30T	5
6	AM30NC	AUX data bus bit 30C	6
33	AM31NT	AUX data bus bit 31T	33
7	AM31NC	AUX data bus bit 31C	7
42	AM32NT	AUX data bus bit 32T	42
43	AM32NC	AUX data bus bit 32C	43
34	AM33NT	AUX data bus bit 33T	34
24	AM33NC	AUX data bus bit 33C	24
46	AM34NT	AUX data bus bit 34T	46
35	AM34NC	AUX data bus bit 34C	35
56	AM35NT	AUX data bus bit 35T	56
57	AM35NC	AUX data bus bit 35C	57
52	AM09NT	AUX data bus bit 09T	52
54	AM09NC	AUX data bus bit 09C	54
69	AADVNT	Advance from AUX T	69
70	AADVNC	Advance from AUX C	70
81	ABSYNT	AUX busy T	81
93	ABSYNC	AUX busy C	93
92	EIMSNT	AUX mainstore inhibit T	92
104	EIMSNC	AUX mainstore inhibit C	104
103	BCPTNT	Bad address parity T	103
114	BCPTNC	Bad address parity C	114
113	MC15NT	AUX select code 15T	112
121	MC15NC	AUX select code 15C	131
115	MC14NT	AUX select code 14T	115
128	MC14NC	AUX select code 14C	128
126	STRNT	Pure store T	126
127	STRNC	Pure store C	127
117	SCSNT	Split cycle store T	117
118	SCSNC	Split cycle store C	118
124	DLCNT	Data load clock T	124
125	DLCNC	Data load clock C	125

TABLE 2-II.- Continued

CPU J2 Pin No.	Signal name	Function	IOP J1 pin No.
107	ASLCNT	AUX SAR load clock T	107
116	ASLCNC	AUX SAR load clock C	116
122	POILNT	Power off interrupt (latched) T	122
123	POILNC	Power off interrupt (latched) C	123
63	AM01NT	AUX data bus bit 01T	63
62	AM01NC	AUX data bus bit 01C	62
74	AM02NT	AUX data bus bit 02T	74
73	AM02NC	AUX data bus bit 02C	73
85	AM03NT	AUX data bus bit 03T	85
84	AM03NC	AUX data bus bit 03C	84
75	AM04NT	AUX data bus bit 04T	75
86	AM04NC	AUX data bus bit 04C	86
97	AM05NT	AUX data bus bit 05T	97
108	AM05NC	AUX data bus bit 05C	108
87	AM06NT	AUX data bus bit 06T	87
98	AM06NC	AUX data bus bit 06C	98
109	AM07NT	AUX data bus bit 07T	109
76	AM07NC	AUX data bus bit 07C	76
64	AM12NT	AUX data bus bit 12T	64
110	AM12NC	AUX data bus bit 12C	110
99	AM13NT	AUX data bus bit 13T	99
111	AM13NC	AUX data bus bit 13C	111
88	AM14NT	AUX data bus bit 14T	88
100	AM14NC	AUX data bus bit 14C	100
112	AM15NT	AUX data bus bit 15T	112
101	AM15NC	AUX data bus bit 15C	101
90	AM19NT	AUX data bus bit 19T	90
102	AM19NC	AUX data bus bit 19C	102
89	AM20NT	AUX data bus bit 20T	89
78	AM20NC	AUX data bus bit 20C	78
91	AM21NT	AUX data bus bit 21T	91
79	AM21NC	AUX data bus bit 21C	79
68	AM22NT	AUX data bus bit 22T	68
80	AM22NC	AUX data bus bit 22C	80
65	AM23NT	AUX data bus bit 23T	65
77	AM23NC	AUX data bus bit 23C	77
58	AM24NT	AUX data bus bit 24T	58
47	AM24NC	AUX data bus bit 24C	47
67	AM25NT	AUX data bus bit 25T	67
55	AM25NC	AUX data bus bit 25C	55
44	AM26NT	AUX data bus bit 26T	44
45	AM26NC	AUX data bus bit 26C	45
23	AM28NT	AUX data bus bit 28T	23
14	AM28NC	AUX data bus bit 28C	14
66	AM18NT	AUX data bus bit 18T	66
32	AM18NC	AUX data bus bit 18C	32
31	AM00NT	AUX data bus bit 00T	31
21	AM00NC	AUX data bus bit 00C	21

TABLE 2-II.- Continued

CPU J5 Pin No.	Signal name	Function	IOP J6 pin No.
53	AM27NT	AUX data bus bit 27T	53
41	AM27NC	AUX data bus bit 27C	41
11	MCM1NT	AUX select code 1T	11
4	MCM1NC	AUX select code 1C	4
30	MCM2NT	AUX select code 2T	30
19	MCM2NC	AUX select code 2C	19

J5 and J6 spares	T	C
1	16	8
2	29	18
3	39	27
4	40	28
5	51	38
6	50	49
7	61	60
8	59	71
9	72	82
10	83	94
11	95	105
12	96	106
13	119	120

TABLE 2-II.- Continued

IOP connector J2 - connector part No. MS27337T24F35PA					
IOP J2 pin No.	Signal name	Connection	IOP J2 pin No.	Signal name	Connection
-1	AGE PR	Slot counter code 00	-40	DB16	Data bus 16
-2	AGE PR	Slot counter code 01	-41	DB17	Data bus 17
-3	AGE PR	Slot counter code 02	-42	DB18	Data bus 18
-4	AGE PR	Slot counter code 03	-43	DB19	Data bus 19
-5	AGE PR	Slot counter code 04	-44	DB20	Data bus 20
-6	PRRET	Return tie 1	-45	DB21	Data bus 21
-7	AGE P1	Clock pulse 01	-46	DB22	Data bus 22
-8	RET2	Return tie 2	-47	DB23	Data bus 23
-9	ROSAR0	ROS address reg 00	-48	RET7	Return tie 7
-10	ROSAR1	ROS address reg 01	-49	DB24	Data bus 24
-11	ROSAR2	ROS address reg 02	-50	DB25	Data bus 25
-12	ROSAR3	ROS address reg 03	-51	DB26	Data bus 26
-13	ROSAR4	ROS address reg 04	-52	DB27	Data bus 27
-14	ROSAR5	ROS address reg 05	-53	DB28	Data bus 28
-15	RET3	Return tie 3	-54	DB29	Data bus 29
-16	ROSAR6	ROS address reg 06	-55	DB30	Data bus 30
-17	ROSAR7	ROS address reg 07	-56	DB31	Data bus 31
-18	ROSAR7	ROS address reg 07	-57	RET8	Return tie 8
-19	ROSAR9	ROS address reg 09	-58	DB32	Data bus 32
-20	ROSAR10	ROS address reg 10	-59	DB33	Data bus 33
-21	RET4	Return tie 4	-60	DB34	Data bus 34
-22	DB00	Data bus 00	-61	DB35	Data bus 35
-23	DB01	Data bus 01	-62	DB36	Data bus 36
-24	DB02	Data bus 02	-63	DB37	Data bus 37
-25	DB03	Data bus 03	-64	DB38	Data bus 38
-26	DB04	Data bus 04	-65	DB39	Data bus 39
-27	DB05	Data bus 05	-66	RET9	Return tie 9
-28	DB06	Data bus 06	-67	DB40	Data bus 40
-29	DB07	Data bus 07	-68	DB41	Data bus 41
-30	RET5	Return tie 5	-69	DB42	Data bus 42
-31	DB08	Data bus 08	-70	DB43	Data bus 43
-32	DB09	Data bus 09	-71	DB44	Data bus 44
-33	DB10	Data bus 10	-72	DB45	Data bus 45
-34	DB11	Data bus 10	-73	DB46	Data bus 46
-35	DB12	Data bus 10	-74	DB47	Data bus 47
-36	DB13	Data bus 10	-75	RET10	Return tie 10
-37	DB14	Data bus 10	-76	DB48	Data bus 48
-38	DB15	Data bus 10	-77	DB49	Data bus 49
-39	RET6	Return tie 6	-78	DB50	Data bus 50

TABLE 2-II.- Continued

IOP connector J2 - connector part No. MS27337T24F35PA					
IOP J2 pin No.	Signal name	Connection	IOP J2 pin No.	Signal name	Connection
-79	DB51	Data bus 51	-104	RDRRD	ROS data reg read
-80	DB52	Data bus 52	-105	RET17	Return tie 17
-81	DB53	Data bus 53	-106	ROSAR	ROS address reg read
-82	RET11	Return tie 11	-107	RET18	Return tie 18
-83	DS4NC	Data sync	-108	RSARL	ROS address reg load
-84	RET12	Return tie 12	-109	RET19	Return tie 19
-85	DMA0	DMA proc 00	-110	CPUDS	DPU data select
-86	DMA1	DMA proc 01	-111	RET20	Return tie 20
-87	DMA2	DMA proc 02	-112	STP1	Stop control 1
-88	DMA3	DMA proc 03	-113	RET21	Return tie 21
-89	DMA4	DMA proc 04	-114	STP2	Stop control 2
-90	RET13	Return tie 13	-115	RET22	Return tie 22
-91	CMWTE	Channel mode write	-116	CNTN	Continue
-92	XMDMA	Channel mode DMA	-117	RET23	Return tie 23
-93	CMINS	Channel mode instr	-118	RST	Reset
-94	CMHW	Channel mode half-word	-119	RET24	Return tie 24
			-120	12 VDC IN	-12 V dc
-95	RET14	Return tie 14	-121	SPARE	
-96	5VDCP	+5 V dc	-122	SPARE	
-97	5VDCN	-5 V dc	-123	SPARE	
-98	12VDCP	+12V dc	-124	SPARE	
-99	REF	Reference	-125	SPARE	
-100	ROSDR	ROS data register	-126	SPARE	
-101	RET15	Return tie 15	-127	SPARE	
-102	RSCLK	ROS clock	-128	SPARE	
-103	RET16	Return tie 16			

TABLE 2-II.- Continued

IOP connector J3 - connector part No. MS27337T24F35PA					
IOP J3 pin No.	Signal name	Function	IOP J3 pin No.	Signal name	Function
1	DO14T	Discrete out 14T	41	DO04T	Discrete out 04T
2	DO14C	Discrete out 14C	42	DO29T	Discrete out 29T
3	DO15T	Discrete out 15T	43	DO20C	Discrete out 20C
4	DO16T	Discrete out 16T	44	DO20T	Discrete out 20T
5	DO17C	Discrete out 17C	45	DO22T	Discrete out 22T
6	DO18C	Discrete out 18C	46	DO28T	Discrete out 28T
7	DO19T	Discrete out 19T	47	DO28C	Discrete out 28C
8	DO12T	Discrete out 12T	48	DO07C	Discrete out 07C
9	DO15C	Discrete out 15C	49	DO06T	Discrete out 06T
10	DO16C	Discrete out 16C	50	DO06C	Discrete out 06C
11	DO17T	Discrete out 17T	51	DO00T	Discrete out 00T
12	DO18T	Discrete out 18T	52	DO04C	Discrete out 04C
13	DO19C	Discrete out 19C	53	DO29C	Discrete out 29C
14	DO21C	Discrete out 21C	54	SPARE	Spare 5C
15	DO10T	Discrete out 10T	55	DO25C	Discrete out 25C
16	DO12C	Discrete out 12C	56	DO25T	Discrete out 25T
17	DO13T	Discrete out 13T	57	DO30C	GPC ID source - C
18	DO01T	Discrete out 01T	58	DO30T	GPC ID source - T
19	DO03C	Discrete out 03C	59	DO05T	Discrete out 05T
20	DO27T	Discrete out 27T	60	DO05C	Discrete out 05C
21	DO26T	Discrete out 26T	61	DO08C	Discrete out 08C
22	DO21T	Discrete out 21T	62	DO08T	Discrete out 08T
23	DO23C	Discrete out 23C	63	DO00C	Discrete out 00C
24	DO23T	Discrete out 23T	64	SPARE	Spare 7C
25	DO09T	Discrete out 09T	65	SPARE	Spare 5T
26	DO10C	Discrete out 10C	66	DI15T	Discrete in 15T
27	DO13C	Discrete out 13C	67	DO31C	GPC-IPL-indicator C
28	DO02T	Discrete out 02T	68	DO31T	GPC-IPL-indicator T
29	DO01C	Discrete out 01C	69	DI17T	Discrete in 17T
30	DO03T	Discrete out 03T	70	DI17C	Discrete in 17C
31	DO27C	Discrete out 27C	71	SPARE	Spare 2C
32	DO26C	Discrete out 26C	72	SPARE	Spare 1T
33	DO22C	Discrete out 22C	73	SPARE	Spare 1C
34	DO24T	Discrete out 24T	74	DI32T	Discrete in 32T
35	DO24C	Discrete out 24C	75	DI32C	Discrete in 32C
36	DO07T	Discrete out 07T	76	SPARE	Spare 7T
37	DO09C	Discrete out 09C	77	DI15C	Discrete in 15C
38	DO11T	Discrete out 11T	78	DI18C	Discrete in 18C
39	DO11C	Discrete out 11C	79	DI18T	Discrete in 18T
40	DO02C	Discrete out 02C	80	DI20T	Discrete in 20T

TABLE 2-II.- Continued

IOP connector J3 - connector part No. MS27337T24F35PA					
IOP J3 pin No.	Signal name	Function	IOP J3 pin No.	Signal name	Function
81	DI19T	Discrete in 19T	105	DI38C	Discrete in 38C
82	SPARE	Spare 2T	106	DI36T	Discrete in 36T
83	DI39C	Discrete in 39C	107	DI34C	Discrete in 34C
84	DI37C	Discrete in 37C	108	SPARE	Spare 6C
85	DI37T	Discrete in 37T	109	DI29C	Discrete in 29C
86	SPARE	SPare 3C	110	+5VDC	+5-volt source (control panels)
87	SPARE	Spare 3T			
88	DI16C	Discrete in 16C	111	DI26T	Discrete in 26T
89	DI16T	Discrete in 16T	112	DI25C	Discrete in 25C
90	DI23T	Discrete in 23T	113	DI24T	Discrete in 24T
91	DI20C	Discrete in 20C	114	DI22T	Discrete ind 22T
92	DI21T	Discrete in 21T	115	DI36C	Discrete in 36C
93	DI19C	Discrete in 19C	116	DI33T	Discrete in 33T
94	DI39T	Discrete in 39T	117	DI31C	Discrete in 31C
95	DI38T	Discrete in 38T	118	DI30T	Discrete in 30T
96	DI35T	Discrete in 35T	119	DI28C	Discrete in 28C
97	DI35C	Discrete in 35C	120	DI27T	Discrete in 27T
98	SPARE	Spare 6T	121	DI24C	Discrete in 24C
99	DI29T	Discrete in 29T	122	DI34T	Discrete in 34T
100	SIGRTN	Control panel signal return	123	DI33C	Discrete in 33C
			124	DI31T	Discrete in 31T
101	DI26C	Discrete in 26C	125	DI30C	Discrete in 30C
102	DI23C	Discrete in 23C	126	DI28T	Discrete in 28T
103	DI22C	Discrete in 22C	127	DI27C	Discrete in 27C
104	DI21C	Discrete in 21C	128	DI25T	Discrete in 25T

TABLE 2-II.- Continued

IOP connector J5 - connector part No. MS27337T24F35PA					
IOP J5 pin No.	Signal name	Function	IOP J5 pin No.	Signal name	Function
1	DI00T	Halt mode CMD	41	DRO09C	GPC N+3 fail vote out
2	DI00C	Halt mode CMD	42	DRO08T	GPC N+2 fail vote out
3	DI01T	Standby mode CMD	43	DRO08C	GPC N+2 fail vote out
4	DI01C	Standby mode CMD	44	DRO07T	GPC N+1 fail vote out
5	DI02T	Run mode CMD	45	DRO07C	GPC N+1 fail vote out
6	DI02C	Run mode CMD	46	DRO00T	GPC fail indicator
7	DI03T	GPC IPL activate CMD	47	DRI04C	GPC N+2 fail vote in
8	DI04T	MM1 IPL select CMD	48	ICB01T	Data bus 01T
9	DI04C	MM1 IPL select CMD	49	ICB01C	Data bus 01C
10	DI05T	MM2 IPL select CMD	50	SPARE	Spare
11	DI05C	MM2 IPL select CMD	51	ICB02T	Data bus 02T
12	DI06T	MM1 ready	52	ICB02C	Data bus 02C
13	DI06C	MM1 ready	53	ICB03T	Data bus 03T
14	DI03C	GPC IPL activate CMD	54	ICB03C	Data bus 03C
15	DI07T	MM2 ready	55	SPARE	Spare
16	DI07C	MM2 ready	56	ICB04T	Data bus 04T
17	DI08T	Discrete in 08T	57	ICB04C	Data bus 04C
18	DI08C	Discrete in 08C	58	DRO00C	GPC fail indicator
19	DI09T	Discrete in 09T	59	ICB05T	Data bus 05T
20	DI09C	Discrete in 09C	60	ICB05C	Data bus 05C
21	DI10T	Discrete in 10T	61	SPARE	Spare
22	DI10C	Discrete in 10C	62	ICB06T	Data bus 06T
23	DI11T	Discrete in 11T	63	ICB06C	Data bus 06C
24	DI11C	Discrete in 11C	64	ICB07T	Data bus 07T
25	DI12T	I/O terminate CMD A	65	ICB07C	Data bus 07C
26	DI12C	I/O terminate CMD A	66	SPARE	Spare
27	DI13T	I/O terminate CMD B	67	ICB08T	Data bus 08T
28	DI13C	I/O terminate CMD B	68	ICB08C	Data bus 08C
29	DI14T	Discrete in 14T	69	SPARE	Spare
30	DI14C	Discrete in 14C	70	SPARE	Spare
31	DRI06T	GPC N+4 fail vote in	71	ICB09T	Data bus 09T
32	DRI06C	GPC N+4 fail vote in	72	ICB09C	Data bus 09C
33	DRI05T	GPC N+3 fail vote in	72	SPARE	Spare
34	DRI05C	GPC N+3 fail vote in	74	ICB10T	Data bus 10T
35	DRI04T	GPC N+2 fail vote in	75	ICB10C	Data bus 10C (FC bus 1)
36	DRI03T	GPC N+1 fail vote in			
37	DRI03C	GPC N+1 fail vote in	76	ICB11T	Data bus 11T (FC bus 2)
38	DRO10T	GPC N+4 fail vote out			
39	DRO10C	GPC N+4 fail vote out			
40	DRO09T	GPC N+3 fail vote out			

TABLE 2-II.- Continued

IOP J5 pin No.	Signal name	Function	IOP J5 pin No.	Signal name	Function
77	ICB11C	Data bus 11C (FC bus 2)	100	ICB19C	Data bus 19C (MM bus 2)
78	SPARE	Spare	101	SPARE	Spare
79	ICB12T	Data bus 12T (FC bus 3)	102	ICB20T	Data bus 20T (PL bus 1)
80	ICB12C	Data bus 12C (FC Bus 3)	103	ICB20C	Data bus 20C (PL bus 1)
81	SPARE	Spare	104	SPARE	Spare
82	ICB13T	Data bus 13T (FC bus 4)	105	ICB21T	Data bus 21T (PL bus 2)
83	ICB13C	Data bus 13C (FC bus 4)	106	ICB21C	Data bus 21C (PL bus 2)
84	SPARE	Spare	107	SPARE	Spare
85	ICB14T	Data bus 14T (FC bus 5)	108	ICB22T	Data bus 22T (LB bus 1)
86	ICB14C	Data bus 14C (FC bus 5)	109	ICB22C	Data bus 22C (LB bus 1)
87	ICB15T	Data bus 15T (FC bus 6)	110	ICB23T	Data bus 23T (LB bus 2)
88	ICB15C	Data bus 15C (FC bus 6)	111	ICB23C	Data bus 23C (LB bus 2)
89	SPARE	Spare	112	SPARE	Spare
90	ICB16T	Data bus 16T (FC bus 7)	113	ICB24T	Data bus 24T
91	ICB16C	Data bus 16C (FC bus 7)	114	ICB24C	Data bus 24C
92	SPARE	Spare	115	SPARE	Spare
93	SPARE	Spare	116	SPARE	Spare
94	ICB17T	Data bus 17T (FC bus 8)	117	SPARE	Spare
95	ICB17C	Data bus 17C (FC bus 8)	118	SPARE	Spare
96	SPARE	Spare	119	SPARE	Spare
97	ICB18T	Data bus 18T (MM bus 1)	120	SPARE	Spare
98	ICB18C	Data bus 18C (MM bus 1)	121	SPARE	Spare
99	ICB19T	Data bus 19T (MM bus 2)	122	SPARE	Spare
			123	SPARE	Spare
			124	SPARE	Spare
			125	SPARE	Spare
			126	SPARE	Spare
			127	SPARE	Spare
			128	SPARE	Spare

TABLE 2-II.- Continued

CPU connector J3 - connector part No. GS25E22-22R2					
IOP J3 pin No.	Signal name	Function	IOP J3 pin No.	Signal name	Function
A	28VP	+28 V dc power	C	SGRND	Safety ground
B	28VN	28 V return	D	SNGRND	Signal ground
CPU connector J7 - connector part No. GS25E22-22R2					
IOP J7 pin No.	Signal name	Function	IOP J7 pin No.	Signal name	Function
A	28VP	+28 V dc power	C	SGRND	Safety ground
B	28VN	28 V return	D	SNGRND	Signal ground
CPU connector J4 - connector part No. MS27337T24F35PA					
IOP J4 pin No.	Signal name	Function	IOP J4 pin No.	Signal name	Function
-1	SPARE		-30	XGBCN	Gate ROS cont C
-2	RET	Return	-31	XLDRDN	Load ROS data reg
-3	XCP2	Clock pulse 2	-32	XBO04N	Bus out bit 04
-4	RET	Return	-33	XBO05N	Bus out bit 05
-5	XCTCLK	Counter clock	-34	XGOSDR	Gate out SDR
-6	RET	Return	-35	RET	Return
-7	SPARE		-36	RET	Return
-8	SPARE		-37	XEXDI	External data
-9	XIADV	Advance	-38	XEDIS	Enable display
-10	XRA04	ROS address bit 4	-39	RET	Return
-11	RET	Return	-40	XHDSC	Hard scan
-12	XRA05	ROS address bit 5	-41	XRA01	ROS address bit 01
-13	XBO00N	Bus out bit 00	-42	XRA02	ROS address bit 02
-14	XBO01N	Bus out bit 01	-43	XBO06N	Bus out bit 06
-15	XIBUSY	Busy	-44	RET	Return
-16	XISEL	Select	-45	XBO07N	Bus out bit 07
-17	RET	Return	-46	XBOC	Bus control
-18	XIREQ	Request	-47	XGDI	Gate data in
-19	XRA06	ROS address bit 06	-48	XEXSS	External start
-20	XRA07	ROS address bit 07	-49	IOPEN	I/O parity error
-21	XBO02N	Bus out bit 02	-50	CPUPEN	CPU parity error
-22	RET	Return	-51	XSTP1N	AGE stop 1
-23	XBO03N	Bus out bit 03	-52	XSTP2N	AGE stop 2
-24	XAEXSS	Allow internal start	-53	XINTN	Inhibit interrupts
-25	XLDCLK	Load clock	-54	RET	Return
-26	XECLKI	External clock	-55	XBUMP	Bump
-27	XRDCLK	Register clock	-56	XBO08N	Bus out bit 08
-28	RET	Return	-57	XWAIT	Wait state
-29	XGBBN	Gate ROS cont B	-58	XMSTL	Monitor stop time

TABLE 2-II.- Concluded

IOP J3 pin No.	Signal name	Function	IOP J3 pin No.	Signal name	Function
-59	GNGTN	Go/no-go time out	-94	RET	Return
-60	RET	Return	-95	XSPERN	Store protect error
-61	XGBAN	Gate ROS cont A	-96	RET	Return
-62	XAION	Stop internal	-97	XAOUTN	AGE data out time
-63	RET	Return	-98	SPARE	
-64	XSYSRN	System reset	-99	SPARE	
-65	XCONT	Continue	-100	XRA10	ROS address bit 10
-66	XRDIN	ROS data in	-101	XRA11	ROS address bit 11
-67	XBO09N	Bus out bit 09	-102	XBO14N	Bus out bit 14
-68	RET	Return	-103	RET	Return
-69	DNUD	Don't use display	-104	XBO15N	Bus out bit 15
-70	RET	Return	-105	XMANTN	Macrointerrupt
-71	SPARE		-106	XMINTN	Microinterrupt
-72	SPARE		-107	RET	Return
-73	SPARE		-108	RET	Return
-74	XAITPN	AGE interrupt	-109	XRA03	ROS address bit 03
-75	XCADVN	Computer advance	-110	XIFDI	Instrofetch data input
-76	XRA00	ROS address bit 00	-111	RET	Return
-77	RET	Return	-112	RET	Return
-78	XRA08	ROS address bit 08	-113	XBO16N	Bus out bit 16
-79	XBO10N	Bus out bit 10	-114	XBO17N	Bus out bit 17
-80	XBO11N	Bus out bit 11	-115	XRPERN	ROS parity error
-81	RET	Return	-116	XRDOT	ROS serial data out
-82	XBO12N	Bus out bit 12	-117	SPARE	
-83	SPARE		-118	SPARE	
-84	RET	Return	-119	XP3	Priority 3
-85	XADEXN	Address exception	-120	RET	Return
-86	RET	Return	-121	RET	Return
-87	XBUSYN	Busy	-122	XP12V	+12V dc
-88	XSATLN	AGE time level	-123	XP5V	+5 V dc
-89	RET	Return	-124	XN5V	-5 V dc
-90	XRA09	ROS address bit 09	-125	RET	Return
-91	SPARE		-126	XP12V	+12 V dc
-92	XBO13N	Bus out bit 13	-127	XBO18N	Bus out bit 18
-93	SPARE		-128	XBO19N	Bus out bit 19

Connector notes:

1. CPU J1 EMXXXX = External data bus

   J2 CBXXXX = Computer address bus for main memory

   J4 = Computer AGE connector

   J5 AMXXXX = Extended memory data bus

2. IOP J1 mates with J2-CPU note

   J2 IOP AGE connector

   J3 discrete inputs and outputs

   J4 mates with J1-CPU note

   J5 discrete inputs/outputs, data bus, and RM discrete I/O

   J6 mates with J5-CPU note

3. C after signal name (i.e., XXP01C) is the complement side and T is the true side.

   N after signal name indicates single-ended.

2.2 CENTRAL PROCESSING UNIT

2.2.1 Overview

The CPU is a digital computer with a processing capability of approximately 400,000 instructions per second based on typical avionics applications. The CPU architecture features over 150 instructions, executed under microprogram control. Arithmetic capabilities include 16- and 32-bit fixed point, and 32-, 39-, and 64-bit floating point operands. The CPU has 14 levels of interrupts, and is capable of detecting over 50 interrupt conditions.

I/O capability consists of a time-shared 32-bit parallel direct I/O, Direct Memory Access (DMA) channel. The CPU memory is organized on an 18-bit halfword basis that includes one storage protect bit and one parity bit. CPU logic is primarily transistor/transistor logic (TTL), and includes medium and large scale integration (MSI, LSI) circuitry.

The CPU can be broken down into several basic elements: main memory; direct I/O, DMA channel; general purpose registers (local store); inner data flow (containing arithmetic/logical unit); interrupt handling circuitry; programmable timers; and fault detection circuitry. Each of these elements is described in the following paragraphs.

2.2.2 Main Memory

The Space Shuttle GPC main storage consists of two separately located but connected core memories. One memory having a storage capacity of 80K words by 36 bits is located in the CPU. The other memory with a storage capacity of 24K words by 36 bits is located in the IOP.

Memory accesses can involve either a halfword (18 bits) or a full word (36 bits). Memory accesses for instructions are always full word; memory accesses for operands are either halfword or full word. Two modes of memory operation are provided: 0.8-μs read/write and a 1-μs read read/modify/write. Access time is 0.4 μs maximum. The above times are increased by 0.1 μs for access to memory located in the IOP. The 0.1 μs increase accounts for cable and buffer amplifier delays between the two LRU's.

The main memory is a 2-1/2 D organized ferrite core memory having a basic modularity of either 8192 words by 18 bits (single density IOP memory) or 16,384 words by 18 bits (double density CPU memory). The 8K (or 16K) pluggable modules are called storage pages. A storage page consists of two parts that unfold, with the 8192-word (or 16,384-word) core array mounted on the inside. The two parts are similar, each containing 8K (or 16K) words by 9 bits of memory.

Two storage pages, together with the timing page, yield a memory of 8K (or 16K) words by 36 bits. The memory is expandable in terms of 36-bit words (pairs of memory pages). The CPU address structure contains 19 address

bits, supporting a memory capacity that is expandable to 256K words by 36 bits.

The storage page is the basic building block of the GPC memory. Ten of the 16K by 18-bit pages are connected by the CPU memory back panel to form an 81,920 by 36-bit storage capacity. Six of the 8K by 18-bit pages in the IOP form the remaining 24,576 by 36-bit words of the GPC memory.

The timing page provides all necessary timing and control signals for operation of the storage pages. Functional elements contained on the page are the timing generator, driver control logic, Storage Address Register (SAR), power switch control logic, and the voltage reference (V_{REF}) circuit.

The memory incorporates power switching to conserve power and to allow unselected (quiescent) main storage pages to remain in a low-power state. The first 36-bit memory access to an unselected pair of memory pages results in a delay of 0.5 μs for execution of the first instruction. Thereafter, main memory access occurs at normal speed. The selected pair of main memory pages is then said to be in the active state. When program execution causes main storage access to take place on a previously unselected pair of pages, the previous pair of pages, which were in the active state, switch to the standby state for 20 μs. This allows instruction execution to take place at normal speeds, should instruction addressing return to the previously selected pair of pages within 20 μs. For the first 20 μs, the previous page pair is said to be in the standby state; thereafter, it switches to the quiescent state.

The main memory is addressable on a halfword basis. A halfword consists of 16 bits. These are the bits that are visible when the accumulator is observed as a result of a short format instruction such as a fixed-point add. The high-order bit is regarded as the sign bit (two's complement notation), and the remaining 15 bits are regarded as the fraction. In the hardware, however, there are 2 additional bits associated with each 16-bit halfword. One of these is a storage protection bit that may be set under program control. If set, this bit protects the associated 16 bits by preventing the 16 bits from being overwritten or modified. If an attempted overwrite occurs, an interrupt is generated. The interrupt indicates that a software flaw exists as a program is attempting to write into a location that contains an instruction or a protected constant. The protect bit can be reset by system software if the location is to be reloaded; e.g., for a program patch or a new initial condition. The instruction which resets the protect bit is privileged and can only be executed in the supervisory machine state.

The second nondata bit associated with each halfword is a parity bit. This bit assigns odd parity to the 16 data bits. Parity errors can cause interrupts to be generated, providing these interrupts are enabled. If the interrupts are masked, parity errors will go undetected and computer execution will proceed as if the parity errors never occurred. (Depending on the problem, nonparity type errors could be annunciated; i.e., illegal op code.)

A full word consists of two halfwords. Full-word instructions are compiled or assembled with the storage protect bits of both halfwords set; however, the setting of a single storage protect bit in either halfword is sufficient to protect the entire full-word instruction.

2.2.3 Direct I/O, Direct Memory Access (DMA) Channel

A 35-bit parallel bidirectional data channel exists between the CPU and IOP. The channel represents the primary communication channel for bidirectional information transfers between the CPU and IOP. When the information transfer is initiated by the CPU, the channel is termed direct I/O and the information transfers are called Program Controlled Output (PCO) or Program Controlled Input (PCI); when the transfer is initiated by the IOP, the channel is called Direct Memory Access (DMA). The 35 bidirectional data lines consist of 32 data bits, 2 parity bits, and 1 storage protect bit.

The CPU controls and monitors the IOP by 32-bit (functional) direct I/O PCO and PCI instructions. A PCO consists of 2 successive 32-bit transfers from the CPU to the IOP. A PCI consists of 2 successive 32-bit transfers also. However, the first occurs from the CPU to the IOP and represents a command to the IOP to transmit the contents of IOP registers back to the CPU. The second 32-bit transfer is then sent from the IOP to the CPU and contains the contents of the IOP registers.

The DMA utilization of the channel provides the IOP with the capability to access main memory for retrieval of IOP program instructions, and for retrieving or placing data being sent to or received from Space Shuttle subsystems into main memory. This use of the channel requires the use of 19 parallel address lines (18 address, 1 parity) from the IOP to the CPU, since the main memory addressing logic physically resides in the CPU. The 18 address bits (functional) allow 2^{18} (262,144) main memory halfword locations to be addressed. The actual size of the memory is 212,992 halfwords.

Since both the CPU and IOP utilize the same memory addressing logic located in the CPU for access to main memory, there is some contention for main memory between the two units. This problem is resolved by two modes of operation. In one mode, the CPU and IOP interleave their memory requests, alternately sharing main memory on essentially an instruction-by-instruction basis. On simultaneous requests, the IOP is given preference. During this mode, the CPU or IOP will repeatedly access main memory if the other processor (IOP or CPU) has no pending memory requests. As soon as the other processor requests a memory access, however, it is given main memory access on the next available cycle. The main memory is thus effectively shared between the two processors. In this mode of operation, the time required for CPU instruction execution is affected by IOP use of the DMA channel. For an IOP which is continuously sharing main memory with the CPU, the effect is to increase the execution time of each CPU instruction by a worst case time of 1.7 μs for each memory access. This comes about as a result of two factors: first, the CPU allows the IOP to access main memory during the execution of a CPU instruction; second, the CPU must wait up to a maximum of

1.7 μs for the IOP to complete its main memory access, before the CPU can again control main memory.

The second mode of operation, called burst mode, allows the IOP exclusive memory access until the number of memory requests in its queue falls below the burst threshold value. This mode is necessitated by the larger number of processors (25) within the IOP, all of which may be concurrently in operation and requiring access to main memory for retrieval of IOP program instructions or for placement of data into main memory. During this mode, the CPU is effectively held off from accessing memory until the IOP relinquishes control of the DMA channel.

2.2.4 CPU Data Formats

Within the CPU, a fixed-point data representation is binary, two's complement, fractional. A halfword is 15 bits plus sign; a full word is 31 bits plus sign.

The floating-point data formats are 32 bits and 64 bits long, differing only in the size of the fraction. The 32-bit length provides sign, 7-bit exponent, and 24-bit fraction. The 64-bit length provides sign, 7-bit exponent, and 56-bit fraction (for some floating point arithmetic operations only 31 bits of the fraction are significant). The 7-bit exponent represents base 16 (hexadecimal) powers, allowing an exponent range of approximately 16^{-64} to 16^{+63}. Floating-point fractions are hexadecimally normalized after every arithmetic operation, a normalized hexadecimal fraction having at least one nonzero bit in the high order four bits of the hexadecimal fraction. The data formats are shown in figure 2-9.

2.2.5 CPU Microprogram

The CPU instruction set includes short and extended precision floating-point, conversion, fixed-point, shifting, and logical instructions. The instructions are implemented in the CPU by a stored set of microprograms.

A microprogram consists of a number of microinstructions that control the hardware in the CPU. Microinstructions control the flow of data and operations to be performed on the data. The microprogram architecture allows each CPU instruction to be executed as a series of microinstructions. Each microinstruction is executed in 0.2 μs except for those microinstructions that access memory. Figure 2-10 illustrates the basic microprogram architecture in the CPU.

The control signals control the flow of data to and from local store (a semiconductor memory used for internal processing) and main store, and selects the inputs to the arithmetic/logical unit.

Each CPU instruction uses a combination of dedicated microinstructions for its particular processing task and common microroutines for instruction

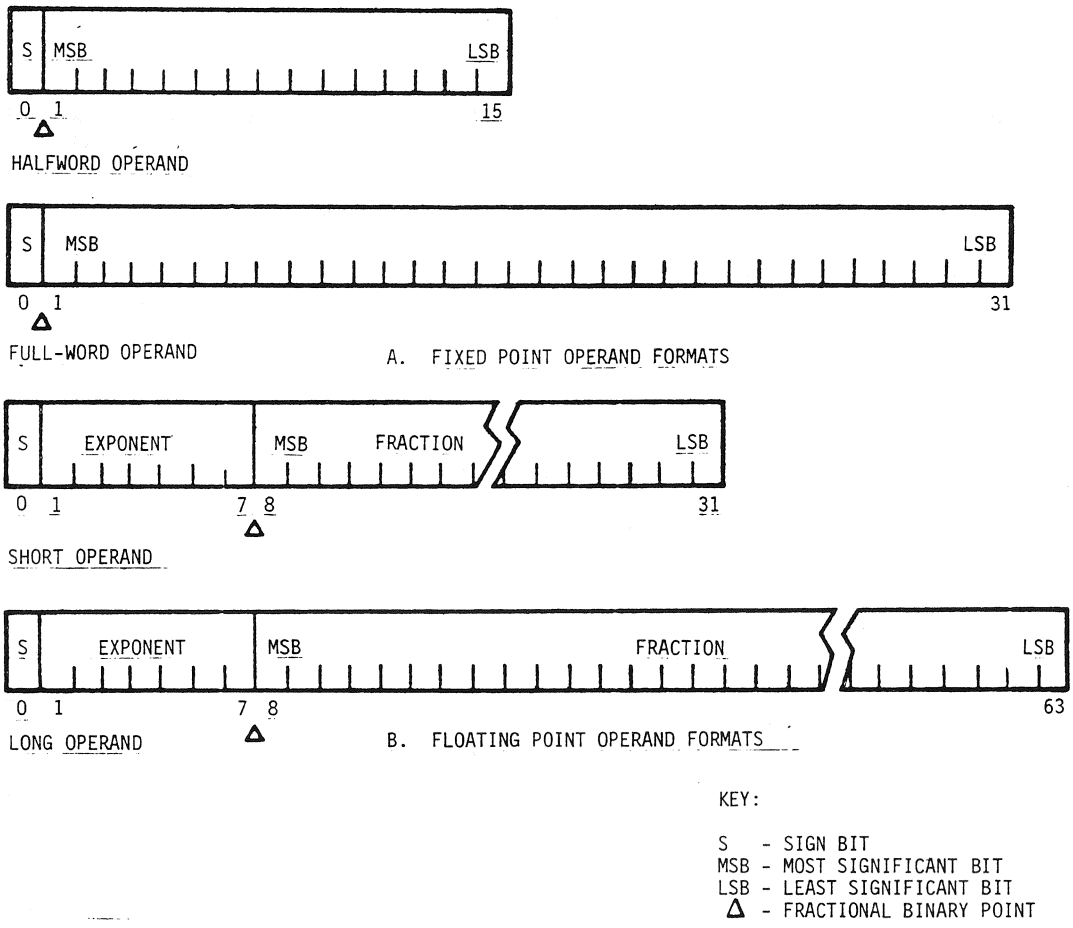


Figure 2-9.- Data formats.

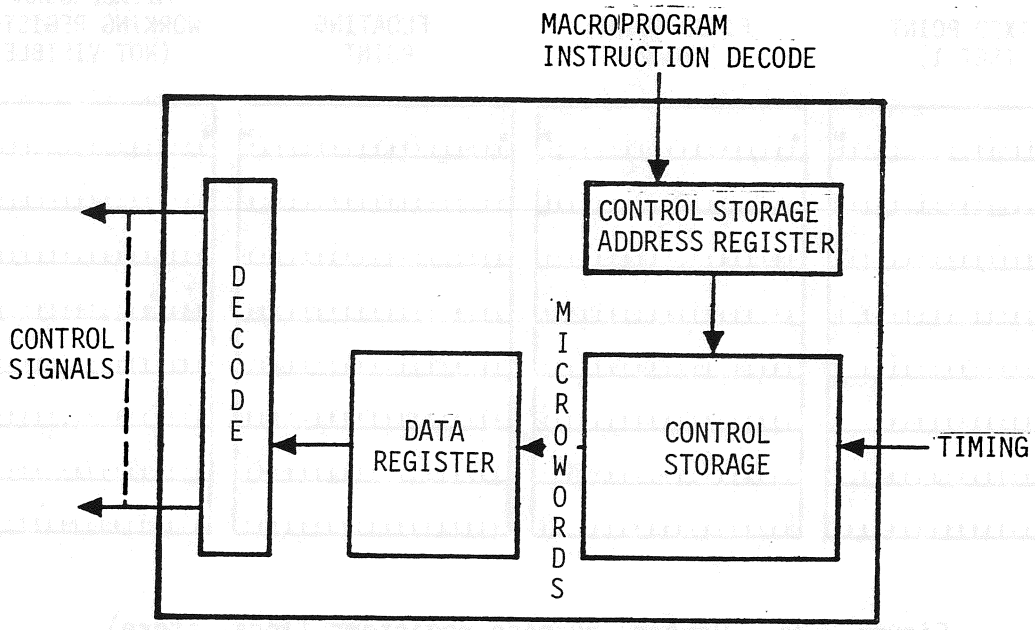


Figure 2-10.- Microprogrammed control structure.

decode, indirect addressing, interrupt handling, and such housekeeping operations as power cycling and self-test.

The microprogram control unit consists of a monolithic microprogram memory, an address register (for addressing the microprogram memory), and a data register (which holds the microinstruction obtained from the microprogram memory). The size of the microprogram memory is 2048 words (microinstructions) by 48 bits.

2.2.6 CPU General Purpose Registers (Local Store)

Arithmetic and logical operations are performed by using sets of programmable (software visible) 32-bit general purpose registers known as local store. There are 32 local store general purpose registers, each 32 bits in length. The registers are arranged into four equal sets of eight registers each. Two sets of the registers are allocated to fixed-point processing and one set is allocated to floating-point processing. The contents of these sets of registers are visible to the programmer via instructions of the CPU set. The remaining set participates in internal CPU processing, and the contents are not visible. Figure 2-11 illustrates the four sets of general purpose registers.

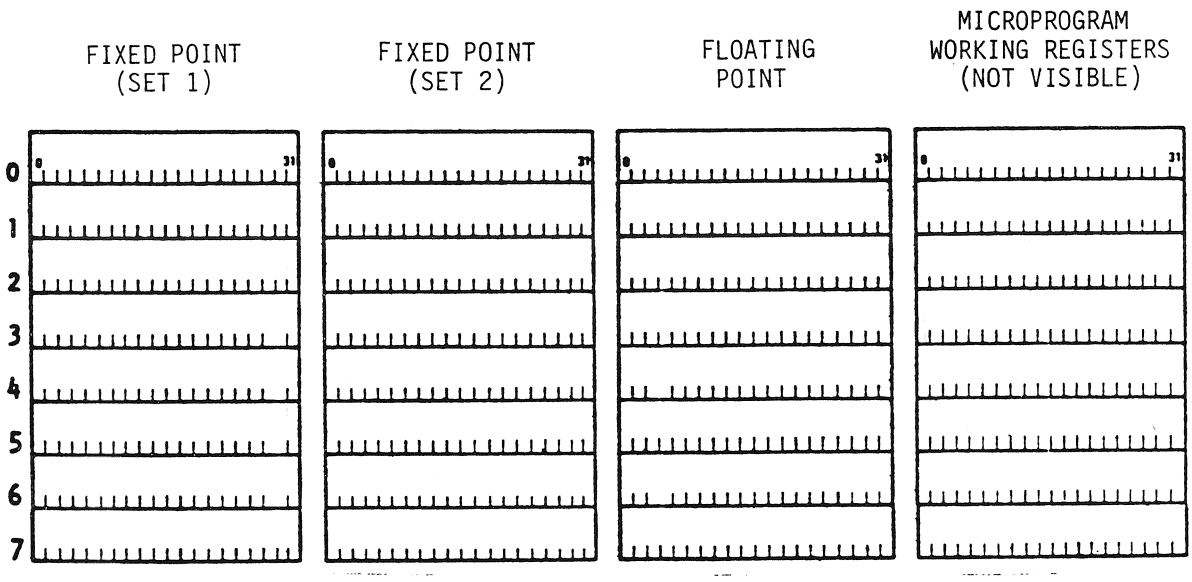


Figure 2-11.- General purpose registers (local store).

At any point in time, the floating-point set and one fixed-point set are active and participate in instruction execution. The registers are explicitly referenced in the CPU instruction set and are used for arithmetic, logical/shifting operations, or as base and index registers for relative addressing.

For some operations, an even/odd pair of general purpose registers can be linked to form a 64-bit double-word register. The most significant half of a double-word operand is contained in the even numbered register while the next higher odd numbered register contains the least significant half. A given set of eight registers is numbered 0 through 7.

2.2.7 CPU Inner Data Flow

Inner data flow consists of the set of general purpose registers (local store), an ALU, and four working registers associated with the ALU. The working registers, designated A, B, C, and D, take part in arithmetic and logical operations. The contents of the registers are not visible to the programmer, and there are no explicit references to the registers in the CPU instruction set.

Figure 2-12 illustrates the logic associated with the inner data flow portion of the CPU.

The local storage cycle and the ALU cycle are synchronized to operate with a 200-nanosecond cycle time. When a local storage output operation is specified, the local storage output is entered into the A register; at the same

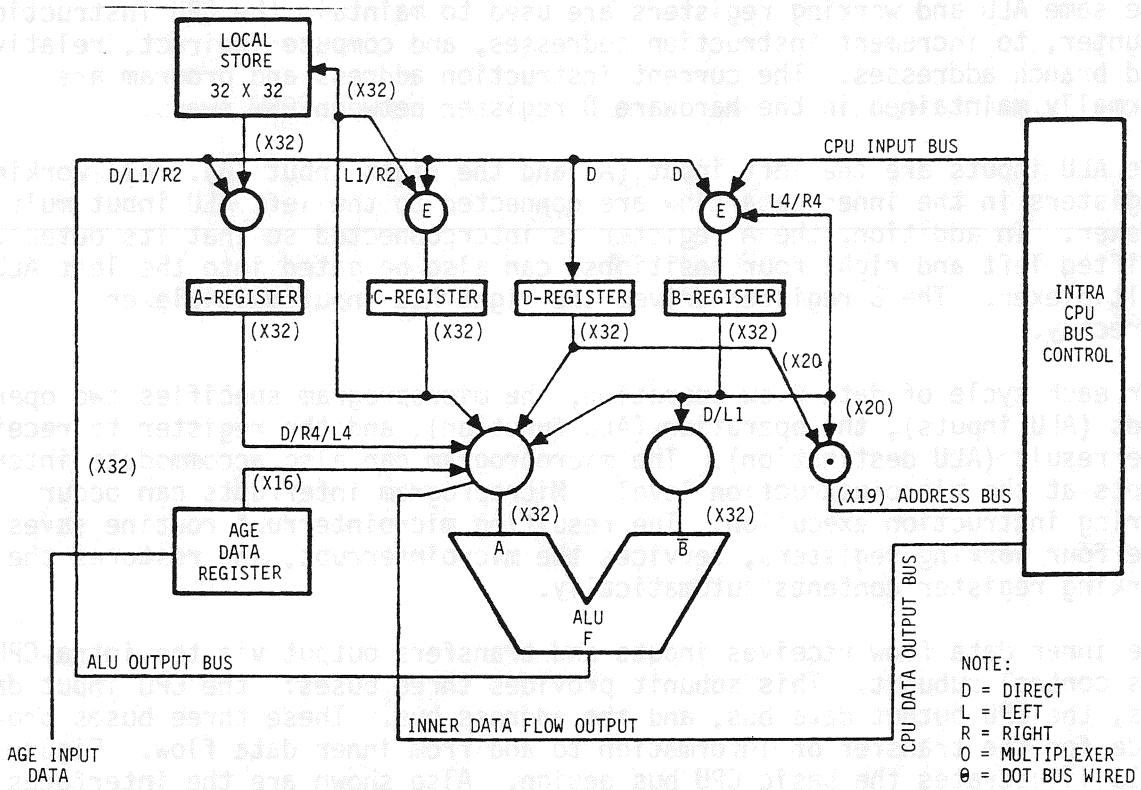


Figure 2-12.- Inner data flow.

time, the ALU output is entered into one of the three remaining working registers, depending upon the CPU instruction being executed. Local storage and the ALU operate synchronously and concurrently under control of separate microprogram control fields. The C register can be input to local store on another 200-nanosecond cycle. The C register operates synchronously and concurrently with the ALU.

The working registers A, B, C, and D are used for temporary working storage during the execution of CPU instructions. The working registers are under microprogram control. During the execution of most instructions, the final result is placed in local store; i.e., in one of the three sets of eight general purpose registers (two fixed-point sets, one floating-point set). Following instruction execution, the contents of the working registers are no longer required and are overwritten during execution of subsequent instructions.

The 32-bit ALU is the focal point for inner data flow operations; it is used to perform all logical and arithmetic computer operations. Simple operations, such as integer add or subtract, require a single pass through the ALU to compute either a 16- or 32-bit result.

The same ALU and working registers are used to maintain the CPU instruction counter, to increment instruction addresses, and compute indirect, relative, and branch addresses. The current instruction address and program are normally maintained in the hardware D register between PSW swaps.

The ALU inputs are the left input (A) and the right input (B). All working registers in the inner data flow are connected to the left ALU input multiplexer. In addition, the A register is interconnected so that its output, shifted left and right four positions, can also be gated into the left ALU multiplexer. The B register drives the right ALU input multiplexer directly.

For each cycle of data flow operation, the microprogram specifies two operands (ALU inputs), the operation (ALU function), and the register to receive the result (ALU destination). The microprogram can also accommodate interrupts at the microinstruction level. Microprogram interrupts can occur during instruction execution. The resulting microinterrupt routine saves the four working registers, services the microinterrupt, and restores the working register contents automatically.

The inner data flow receives inputs and transfers output via the intra-CPU bus control subunit. This subunit provides three buses: the CPU input data bus, the CPU output data bus, and the address bus. These three buses provide for the transfer of information to and from inner data flow. Figure 2-13 illustrates the basic CPU bus design. Also shown are the interfaces to the IOP channel controller and to the 24K x 36 bits of main memory located in the IOP.

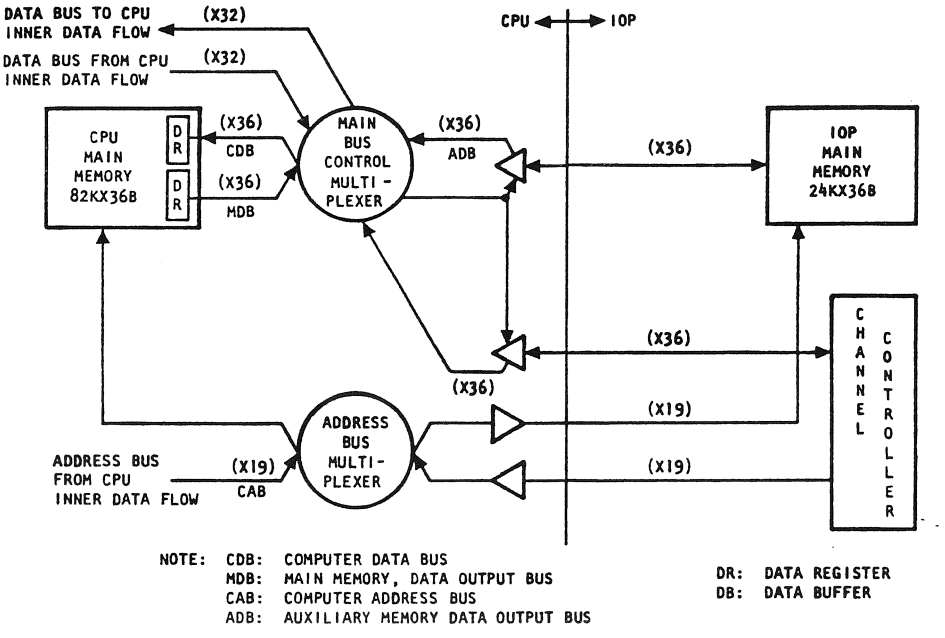


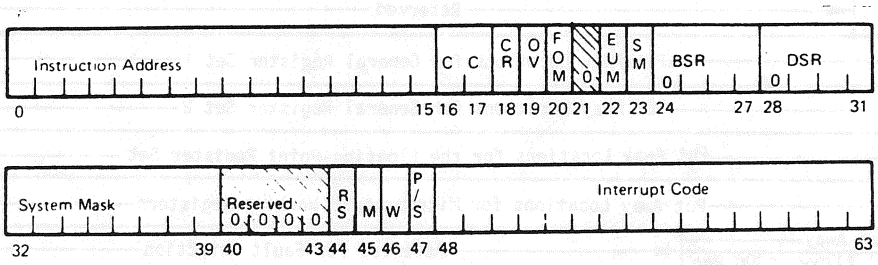
Figure 2-13.- CPU bus control.

2.2.8 Interrupts

An interrupt is a condition which occurs in the GPC hardware or software which causes the GPC to stop, or "interrupt," its current processing, and to execute a special program that is capable of responding to the condition. The condition which causes the interrupt might be a hardware error, such as a parity error; a software error, such as divide by zero; or it might be the result of something that isn't an error at all, such as a supervisory call. It is important to remember that not all interrupts are errors; some interrupts occur as a result of the GPC's normal operation.

When an interrupt is sensed by the CPU, a sequence of events automatically takes place. The PSW (shown in figure 2-14) currently being used is placed in main storage, and a new PSW from main storage is placed in the PSW register. The PSW is used to control instruction sequencing and to hold and indicate the status of the system in relation to the program presently being executed.

The current PSW contains the address of the next CPU instruction that is to be executed. Therefore, when a new PSW is loaded into the PSW register, it is used to address the first instruction of the new processing sequence. After some interrupts, the old PSW can be loaded, and processing of the interrupted software can continue. However, due to the nature of some interrupts, this is not always possible.



Bits

0-15	Next Instruction Address	
16-17	Condition Code	
18	Carry Indicator	
19	Overflow Indicator	
20	Fixed Point Arithmetic Overflow Mask	
21	Reserved	
22	Floating Point Exponent Underflow Mask	
23	Significance Mask	
24-27	Branch Sector Register	
28-31	Data Sector Register	
32	Counter 1 Mask	System Mask
33	Counter 2 Mask	
34	Instruction Monitor Mask	
35	External Interrupt 0 Mask	
36	External Interrupt 1 Mask	System Mask
37	External Interrupt 2 Mask	
38	External Interrupt 3 Mask	
39	External Interrupt 4 Mask	
40-43	Reserved	
44	Register Set (GR set 0 or 1)	
45	Machine Check Mask	
46	Wait State Bit (Wait/Process)	
47	Problem/Supervisor State Control Bit	
48-63	Interrupt Code for Program Machine Check and Special External Interrupts	

Figure 5-14.- The PSW fields.

There are 14 PSW main storage locations, and as far as the CPU hardware is concerned, only 14 interrupts exist. The 14 interrupts are designated as levels. Thus, the CPU hardware processing consists of decoding which interrupt level has occurred, and swapping the present PSW for the appropriate PSW in main memory. Figure 2-15 shows the storage locations for the new and old PSWs.

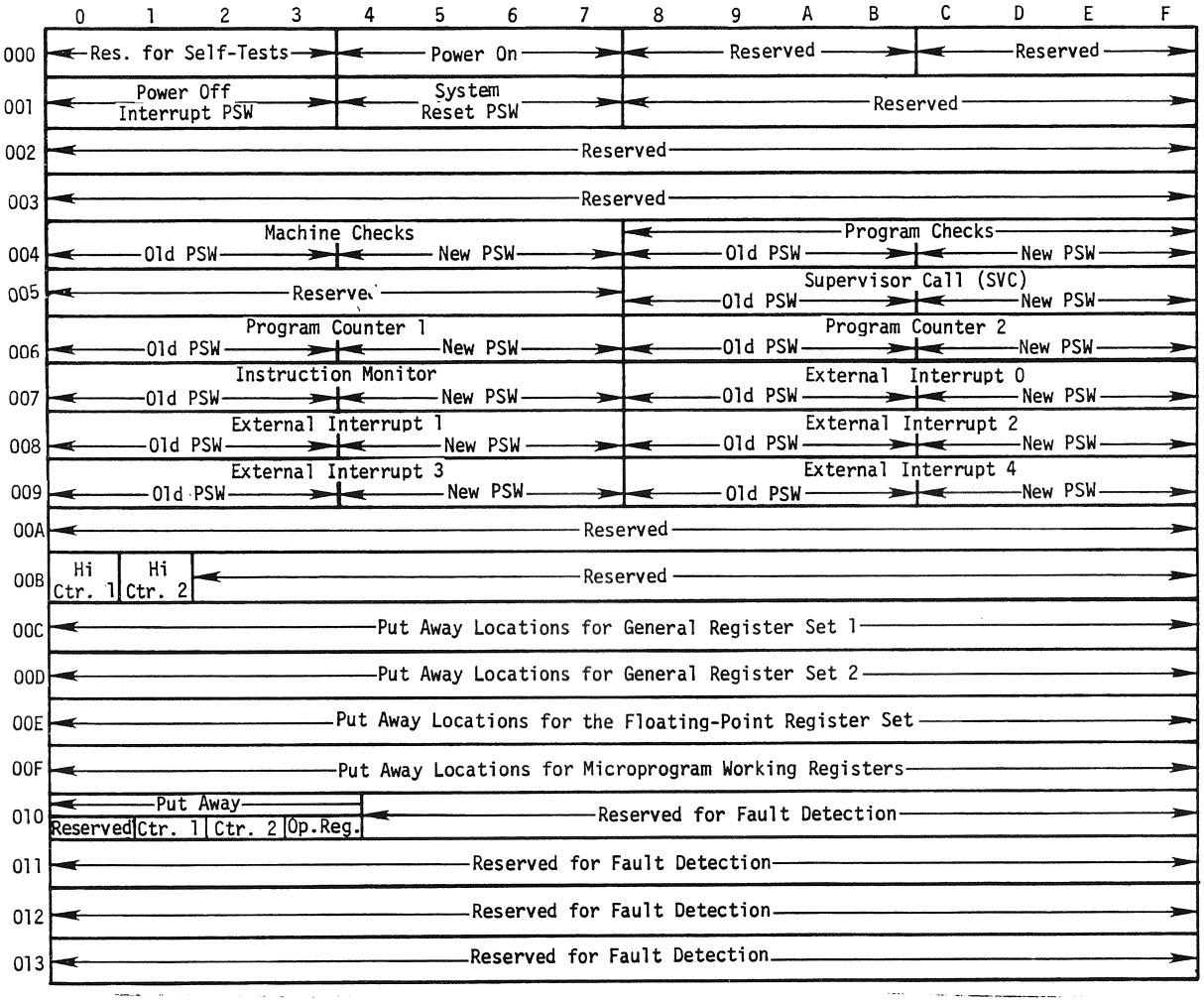


Figure 2-15.- Preferred storage area assignments.

The 14 interrupt signals are shared by a total of 44 different interrupt conditions. Of the 14 interrupt signals, nine are internal to the CPU and five are IOP to CPU interrupts. Since 44 interrupt conditions can cause 14 interrupt signals to be generated, provisions have been made to allow the CPU to determine which interrupt condition led to the generation of an interrupt. For interrupts generated inside the CPU, an interrupt code is provided in the PSW. By reading this code, it can be determined which interrupt condition led to the generation of the interrupt. For IOP-related interrupts, similar information is stored in registers in the IOP and can be obtained via a PCI instruction.

NOTE: All of the possible interrupts are not used.

44 interrupts:

No.	Name	No.	Name
1	PWR ON OR AGE	23	PROGRAM COUNTER #1
2	RESET	24	PROGRAM COUNTER #2
3	POWER OFF	25	EXTERNAL #0 (IOP)
4	MAIN STORAGE PARITY ERR (CPU)	26	EXTERNAL #1 (IOP)
5	MAIN STORAGE PARITY ERR (DMA)	27	DMA - IOP ADDRESS
6	EXT STOR ADDR PARITY	28	I/O STORE PROTECT
7	EXT STOR DATA PARITY	29	I/O WRITE DATA
8	ADDR OUTSIDE MEMORY	30	PCI DATA PARITY
9	STORE PROTECT VIOLATION	31	I/O ADDRESS PARITY
10	ROS PARITY ERROR	32	AGE INTERRUPT
11	INSTRUCTION MONITOR	33	EXTERNAL #2 (IOP)
12	ILLEGAL OPERATION	34	EXTERNAL #3 (IOP)
13	PRIVILEGED INSTRUCTION	35	EXTERNAL #4 (IOP)
14	FIXED POINT OVERFLOW	36	
15	SIGNIFICANCE ERROR	37	
16	EXPONENT UNDERFLOW	38	
17	EXPONENT OVERFLOW (CVFX)	39	NOT USED
18	EXPONENT OVERFLOW (F.P.)	40	
19	DIVIDE-BY-ZERO	41	
20	UNNORMALIZED DIVIDE INPUTS	42	
21	SUPERVISORY CALL	43	
22	INITIATE PUTAWAY	44	

Table 2-III is a list of all the possible interrupt conditions along with other information that describes these interrupts. The information was obtained from Software Awareness Memo Number 28, entitled "Software Error Conditions/Handling Baseline" and the Backup System Services PRD (MG038101). What follows is an explanation of the columns in the table.

1. **No.** — The number of the interrupt condition.

2. **Name** — A descriptive name.

   NOTE: Interrupts 25 and 26, which are generated in the IOP, are named External 0 and External 1, respectively. The additional names included for these interrupts are the specific causes in the IOP for the generation of the interrupt. As stated previously, the CPU determines these causes by using a PCI to read registers in the IOP where this information is stored.

3. **Priority** — Indicates the hardware priority number. Simultaneous interrupts are serviced in order of priority. The smaller the hardware priority number, the higher the priority. Interrupts having the same priority are mutually exclusive.

4. **Maskable** — Some interrupts can be masked by setting a bit in the PSW. If an interrupt is masked, it means it will be ignored. A "No" in this column means the interrupt cannot be masked; a "Yes" means the interrupt can be masked.

5. **Remain pending** — This column indicates whether a masked-off interrupt is held pending until unmasked (Yes), or simply disappears as if it had never occurred (No).

6. **PSW mask bit** — This column indicates which bit in the PSW can be set to mask the interrupt, if it is maskable.

7.	PASS initialized	Indicates in what state the interrupt is initialized for FCOS and application's (appl) software. "E" indicates that the interrupt is enabled (not masked); "D" indicates that the interrupt is disabled (masked).
8.	Hardware/software	Indicates whether the error was generated by hardware or software.
9.	Classification	There are five classes of interrupts.
	a. Power (P)	This interrupt occurs when primary power is removed from the system for any reason. The current PSW, the general register set 1 and 2, and the operational registers are put away (stored) in main storage for future reference.
	b. Machine Check (M)	When not masked, this interrupt class occurs following the detection of malfunction. The current instruction is then terminated and the interrupt taken. The CPU goes to software wait state (PASS only). When masked, the interrupt does not remain pending.
	c. Program (PG)	This class of interrupt arises from improper specification or use of instructions or data. When masked, program interrupts do not remain pending. When invalid instruction or address detection is provided, the resulting program interrupts cannot be masked.
	d. Supervisor Call (SVC)	This interrupt results from the execution of the SVC instruction. This instruction can be used to switch from the problem to the supervisor state.
	e. System (SY)	This class of interrupt results from program counter timeouts, and from conditions outside the CPU. Provisions are made for seven interrupt levels, two for the program counters and five for the IOP generated interrupts.

10. Present action

This column indicates the action taken by application and FCOS software for each interrupt. A WAIT indicates that the CPU goes to software halt. Forced close indicates that the processing for the particular application program running when the interrupt is received is terminated. However, the application program will be executed again during the next cycle for which it is scheduled. When a PASS GPC force closes a process, the fault message "GPC BITE 1(2,3,4,5)" is annunciated. DHPTRR indicates that the FCOS will Dispatch Highest Priority Task Ready to Run. The other terms used should be self-explanatory.

11. BFS action

This column indicates the action taken by the BFS for each interrupt. A "restart" in this column indicates that the BFS will attempt to restart processing by performing I/O initialization and minor cycle preparation. For a software-initiated restart, the BFS will annunciate "GPC BITE". For a power transient restart, "GPC PWR" will be annunciated.

For "error trap" recovery, the BFS determines whether or not a dispatched job was executing when the error occurred. If a dispatched job was executing, the control is transferred to the job's dispatch entry point minus one full word. If a dispatched job was not executing, a restart is initiated. For error trap recovery marked with an "*," bit 38 of the interrupted PSW is checked, and if it is 1, the recovery is performed as defined above. If bit 38 is 0, the overflow bit in the interrupted PSW is set, and execution of the interrupted process continues.

12. Notes

These notes are given as an aid in understanding the interrupt.

NOTE: If a GPC is still functioning after an interrupt resulting from a hardware or software problem is processed, the GPC will log the error in the GPC Error Log which is downlisted. For information on interpreting the GPC Error Logs, see PASS GPC Error Monitoring (SCP 1.9) and BFS GPC Error Monitoring (SCP 1.13) in the DPS Console Handbook.

TABLE 2-III.- INTERRUPT CONDITIONS

No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
1	Power ON or AGE	1	No	-	-	-	-	Hardware	P	Respond[d]	Respond[d]	e	Initiated by power supply
2	Reset	1	No	-	-	-	-	Hardware	P	Respond[d]	Respond[d]	e	Initiated by power supply
3	Power off	1	No	-	-	-	-	Hardware	P	Respond[d]	Respond[d]	e	Initiated by power supply
4	Main storage (low) parity error (CPU)	2	Yes	No	45	E	E	Hardware	M	WAIT[c]	WAIT	Error trap	Hardware error; imprecise PSW; initiated by CPU BITE; read operation; bad parity restored; instruction or data
5	Main storage (low) parity error (DMA)	3	Yes	No	45	E	E	Hardware	M	WAIT	WAIT	Error trap	Hardware error; imprecise PSW; initiated by CPU BITE; read operation; bad parity restored; instruction or data
6	Extended storage address parity (CPU and DMA)	4	Yes	No	45	E	E	Hardware	M	WAIT	WAIT	Error trap	Hardware error; imprecise PSW; initiated by CPU BITE; read or write operation aborted
7	Extended storage data parity (CPU and DMA)	5	Yes	No	45	E	E	Hardware	M	WAIT	WAIT	Error trap	Hardware error; imprecise PSW; initiated by CPU BITE; read or write operation aborted

[a]E = Enable; D = Disable.
[b]P = Power; M = Machine Check; PG = Program; SVC = Supervisory Call; SY = System
[c]WAIT = CPU in WAIT state.
[d]PASS GPC not usable if powered on or off while under S/W control (MODE SW in STBY or RUN).
[e]BFS may be used if powered on while under S/W control. BFS will RESTART for power transient

TABLE 2-III.- Continued

No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
8	Address outside memory (CPU)	6	No	-	-	-	-	Software	PG	Forced close	DHPTRR[f]	Error trap	Imprecise PSW; initiated by software; terminates
9	Store protect violation (CPU)	7	No	-	-	-	-	Software	PG	Forced close	DHPTRR[f]	Error trap	Imprecise PSW; initiated by software; terminates
10	ROS parity error (CPU)	8	Yes	No	45	E	E	Hardware	M	WAIT	WAIT	Error trap	Imprecise PSW; initiated by CPU BITE; read; memory contents are unchanged
11	Instruction monitor (no store protect bit)	9	Yes	No	34	E	E	Software	PG	Forced close	DHPTRR[f]	Error trap	Imprecise PSW; initiated by software;terminates
12	Illegal operation	10	No	-	-	-	-	Software	PG	Forced close	DHPTRR[f]	Error trap	Imprecise PSW; undefined instruction; initiated by software; no operation is performed
13	Priviledged operation[g]	10	No	-	-	-	-	Software	PG	Forced close	-	Error trap	Supv. state instruction brought in for execution in problem state; initiated by software; not executed

[a]E = Enable; D = Disable.
[b]P = Power; M = Machine Check; PG = Program; SVC = Supervisory Call; SY = System
[f]DHPTRR = Dispatch Highest Priority Task Ready to Run; set "MSC interruptable" flag
[g]Insert storage protection bit; load PSW; set system mask (PSW); Program Controlled Output (PCO); internal control..

TABLE 2-III.- Continued

No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
14	Fixed-point overflow	10	Yes	No	20	D	D	Software	PG	Masked	Masked	Masked	Programming error interrupt; initiated by software; actual products in registers (add, sub); registers unchanged (divide)
15	Significance (F.P. fraction = 0) (exp. ≠ 0)	10	Yes	No	23	D	D	Software	PG	Masked	Masked	Masked	Programming error interrupt; initiated by software; true zero result is forced in operation when interrupt is masked
16	Exponent underflow (F.P.) (exponent < 0, fraction ≠ 0)	10	Yes	No	22	D	D	Software	PG	Masked	Masked	Masked	Programming error interrupt; initiated by software; exponent <0, fraction = 0; actual products in registers (add, sub); register unchanged (mult. and div.); true zero result is forced in operation when interrupt is masked
17	Exponent overflow (convert) CVFX	10	No	-	-	-	-	Software	PG	Log and ignore	Log and ignore	Error trap*	Iniated by software; terminated; registers unchanged

[a]E = Enable; D = Disable.

[b]P = Power; M = Machine Check; PG = Program, SVC = Supervisory Call; SY = System.

TABLE 2-III.- Continued

No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
18	Exponent overflow (F.P.)	10	No	·	·	·	·	Software	PG	Log and ignore	Log and ignore	Error trap	Programming error interrupt; initiated by software; terminated; characteristic exceeds 127; registers,unchanged
19	Divide by zero (F.P.)	10	No	·	·	·	·	Software	PG	Log and ignore	Log and ignore	Error trap	Programming error interrupt; initiated by software;operation suppressed; registers and storage unchanged
20	Unnormalized divide inputs (F.P.)	10	No	·	·	·	·	Software	PG	Log and ignore	Log and ignore	Error trap	Programming error interrupt; initiated by software; completes; not generated if exponent overflow is set
21	Supervisory call	11	No	·	·	·	·	Software	SVC	Respond	Respond	Respond	Normal program interrupt; initiated by control hardware
22	Initiate putaway	12	No	·	·	·	·	Software	P	Respond	Respond	Respond	Receive on power down; initiated by power supply
23	Program counter #1	13	Yes	Yes	32	E	E	Software	SY	Respond	Respond	Respond	Normal program interrupt, 32-bit interval times; initiated by control hardware

[a]E = Enable; D = Disable.
[b]P = Power; M = Machine Check; PG = Program; SVC = Supervisory Call; SY = System

TABLE 2-III.- Continued

No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
24	Program counter #2	14	Yes	Yes	33	D	E	Software	SY	Respond	Masked	Respond	Normal program interrupt; 32-bit interval timer; initiated by control hardware
25	External #0 (IOP)	15	Yes	Yes	35	D	E	Software	SY	-	Masked	-	Level A interrupt; initiated by control hardware
	Watchdog timer									WAIT/ ignore [h]	Masked	Log and ignore	Termination control should be re-evaluated
	IOP fail votes									WAIT/ ignore [h]	Masked	Log and ignore	Termination control should be re-evaluated
	IOP reset									Ignore	Masked	Restart	IOP response to HALT; power ON; or PCO master RESET
	IOP micro-program ROS parity									WAIT/[c] RESET IOP	Masked	Restart	Automatic RESET by hardware
	IOP clock failure									WAIT/RESET IOP	Masked	Restart	Automatic RESET by hardware
	Spare									WAIT/RESET IOP	Masked		
26	External #1 (IOP)	16	Yes	Yes	36	D	E	Software	SY	-	Masked	-	Level B interrupt; initiated by control hardware

[a] E = Enable; D = Disable.
[b] P = Power; M = Machine Check; PG = Program; SVC = Supervisory Call; SY = System.
[c] WAIT = CPU in WAIT state.
[h] If termination control latch is enabled, CPU WAIT state is commanded; if latch is disabled, interrupt is ignored. (Not used in PASS.)

TABLE 2-III.- Continued

No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
26	PC I/O IOP parity									WAIT/RESET IOP	Masked	Restart	Date or command
	DMA instruc-tion read parity									WAIT/RESET IOP	Masked	Restart	
	DMA data read parity									WAIT/RESET IOP	Masked	Restart	
	DMA burst									WAIT/RESET IOP	Masked	-	Not wired in production computers
	DMA queue overflow									WAIT/RESET IOP	Masked	Restart	
	DMA timeout									WAIT/RESET IOP	Masked	Restart	DMA timeout does not occur for DMA of addresses outside of memory
27	(DMA) IOP address spec specification[i]	16	Yes	Yes	45	D	E	Software	SY	Forced closed	DHPTRR[f]	Restart	DMA address outside memory, initiated by control hardware (CPU); storage is unmodified; imprecise PSW
28	I/O store protect (DMA)[i]	16	Yes	Yes	45	D	E	Software	SY	Forced closed	DHPTRR	Restart	Caused by a program error (CPU); storage is unmodified; imprecise PSW

[a] E = Enable; D = Disable.
[b] P = Power; M = Machine Check; PG = Program; SVC = Supervisory Call; SY = System.
[f] DHPTRR = Dispatch Highest Priority Task Ready to Run; set "MSC interruptable" flag.
[i] External #1 interrupt is generated by microcode handling this interrupt.

TABLE 2-III.- Concluded

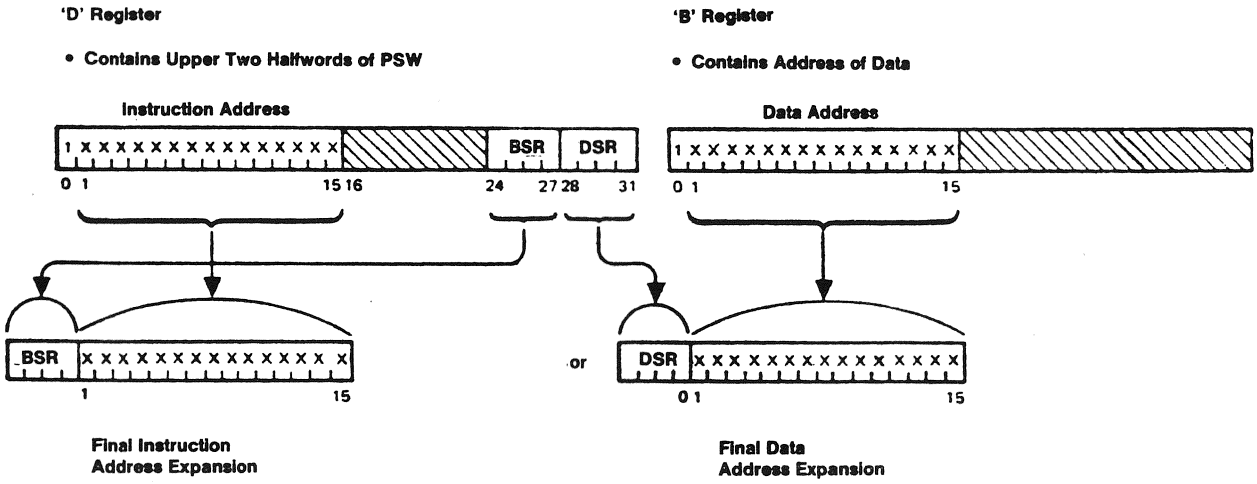
No.	Name	Priority	Mask-able	Remain pending	PSW mask bit	PASS initialized (a)		Hardware/ software	Classi-fication (b)	PASS present action		BFS action	Notes
						FCOS	Appl			Appli-cation	FCOS		
29	I/O write (data parity DMA in)[i]	16	Yes	Yes	45	D	E	Hardware	SY	WAIT/[c] RESET IOP	Masked	Restart	Bad data word; initiated by control hardware (CPU); terminated; storage is unmodified
30	PCI data parity[i]	16	Yes	Yes	45	D	E	Hardware	SY	WAIT/RESET IOP	Masked	Restart	Bad data word; initiated by control hardware (CPU)
31	I/O address parity (DMA) [i]	16	Yes	Yes	45	D	E	Hardware	SY	WAIT/RESET IOP	Masked	Restart	Bad address; initiated by control hardware; storage is unmodified
32	AGE interrupt[i]	16	Yes[j]	Yes	36	D	E	Software	SY	(k)	Masked	Log and ignore	N/A to flight software; external interrupt; initiated by control hardware; first interrupt is transparent to the programmer
33	External #2 (IOP)	17	Yes	Yes	37	D	E	Software	SY	Respond	Masked	-	Programmed interrupt; level C; initiated by control hardware
34	External #3 (IOP)	18	Yes	Yes	38	D	E	Software	SY	Masked	Masked	-	Spare; initiated by control hardware
35	External #4 (IOP)	19	Yes	Yes	39	D	E	Software	SY	Masked	Masked	-	Spare; initiated by control hardware

[a] E = Enable; D = Disable.
[b] P = Power; M = Machine Check; PG = Program; SVC = Supervisory Call; SY = System
[c] WAIT = CPU in WAIT state.
[i] External #1 interrupt is generated by microcoda handling this interrupt.
[j] Maskable only with External #1 interrupt mask.
[k] If I/O is active, ignore interrupt; if I/O is quiescent, go to effective WAIT state.

2.2.9 CPU Extended Addressing

The Space Shuttle GPCs contain 104K full words of main store (208K half-words). This requires the use of 18 address bits. The DMA channel between the IOP and CPU contains 18 address bits. The internal CPU address bus structure contains 19 address bits, allowing a maximum expansion of main memory to 256K full words (512K halfwords).

Addresses beyond 15 bits require the use of BSR and DSR fields within the current PSW. Nineteen-bit addresses are formed as shown in the following sketch (the maximum address used in the existing GPC is 18 bits). The upper half of the current PSW (which contains the address of the next instruction that is to be executed) is held within the D register of the ALU. The 19-bit address for the acquisition of operands is held within the B register of the ALU.



2.2.10 CPU Real-time Clock Timers

There are two programmable real-time clock timers (downcounters) in the CPU. The timers have a LSB of 1 microsecond. Values of FFFF FFFF through 0000 0000 can be inserted into either timer for a maximum time interval of 65.536 milliseconds. When a timer counts down through the value of zero, an internal CPU interrupt is generated. The application programs in the CPU can thus be interrupted at predetermined elapsed time intervals.

Each of the two timers is individually maskable by a bit in the PSW. This bit also determines the timer mode of operation.

Each timer interrupt is 32 bits long, with 16 bits (high order) residing in main storage and 16 bits (low order) in a hardware downcounter in the CPU. The low-order half of the timer decrements continuously.

If the timer is unmasked when the low-order half of the timer passes through zero, a test is performed on the high-order software bits. If all bits are zero, an interrupt is generated and the software half is decremented. Otherwise, the software half is merely decremented. The process continues until an interrupt is generated. After generating the interrupt, the timer continues to function as a full 32-bit decrementing timer.

If the timer is in the masked state when the low-order half of the timer reaches zero, the software half of the timer is neither tested nor decremented. A flag is set instead. When the timer is eventually unmasked (must be within 65.356 ms), the presence of this flag causes an immediate test to be performed on the software half of the timer. If the software bits are zero, an interrupt is generated and the software half is decremented. At this point, the timer has returned to the status of a full 32-bit decrementing counter. Reloading the timer (with any value) before unmasking it always causes the aforementioned flag to be reset. A timer interrupt thus can always be averted by the single action of reloading the timer.

A third timer exists in the GPC and physically resides in the IOP. The function of this timer is to detect an improperly cycling CPU. It is designated the "watchdog" or go/no-go timer. The intent is that the timer will be periodically reloaded by the CPU (PCO command) before it ever times out. If a timeout occurs, an interrupt will be sent to the CPU to enable error recovery to take place. The interrupt is maskable by a bit in the PSW. The interrupt will remain pending if it occurs when masked.

The operation and use of the "watchdog" timer is described further in this section under IOP Redundancy Management Logic.

2.2.11 CPU Fault Detection

A fault detection capability exists in the CPU utilizing BITE. The fault detection capability, when used with an appropriate Self-Test Program (STP), is designed to provide greater than 95-percent detection of the hardware failures that affect computer operation. The BITE hardware features are described below.

A. Main store parity

   All data and instruction transfers from main storage are checked for odd parity on a halfword basis. Selected data words with even parity are resident in main storage for use by the STP to test the parity checking circuits. Main storage parity errors are distinguishable in terms of whether or not they occurred as the result of a CPU or IOP memory access.

B. I/O parity assignment and check

   Odd parity is generated and checked on all I/O transfers on the direct I/O, DMA channel.

C. Microstore parity

Odd parity is checked on all CPU microstore read operations.

D. Power monitoring circuits

The power supply output voltages are monitored for loss of voltage or out-of-tolerance dc voltages. If this occurs, the fault indicator is automatically set.

E. Storage protection

Storage protection prevents hardware or software overwriting of specified data and instruction locations. An interrupt is generated when an attempt is made to alter a protected location. Locations for protection are specified either during initial load or under control of the program.

F. Instruction monitor

The instruction monitor prevents the mistaken execution of variable data as an instruction. If it detects the attempted execution of a 16- or 32-bit data field (as indicated by the absence of a storage protect bit), an interrupt is generated.

G. Illegal operation

Instruction operation codes are tested as each is executed. Certain op codes defined as illegal will result in a program-class interrupt if executed. The intent is to minimize problems due to executing data as instructions.

H. Illegal address

An attempt to address main storage outside the configured number of words results in an address specification error. A program-level interrupt is generated.

2.2.12 CPU Packaging Design

The main structural member is a dip-brazed aluminum alloy housing with finned heat exchangers on the sidewalls. Forced cooling air is passed through and contained in the heat exchanger, preventing direct impingement of cooling air on components. Aluminum rails brazed or bolted to the heat exchangers provide mounting surfaces and thermal conduction paths for all subassemblies.

Top and bottom access covers using captive fasteners provide direct access to subassemblies inside. All pluggable electronic modules, power supply, and radio frequency interference filters contain captive fasteners for ease of maintenance.

The CPU is packaged on pluggable subassemblies which, except for main storage, plug into a single Multilayer Interconnection Board (MIB) called a back panel. The back-panel MIB contains all interconnecting buses and control lines. A separate MIB back panel is used for the main storage subassemblies.

The power distribution wiring harnesses in the CPU are the only interconnection devices using discrete wiring. The power distribution wires are terminated at one end in crimped contacts for insertion into a connector; the opposite end is terminated into plated-through holes in MIBs. The wires are strain relieved via potting wells on the back panel and an insulated clamp on the page frame.

The basic electronic module is a pluggable subassembly called a page. Most logic pages employ two MIBs bonded to a metal frame, with insulators separating the boards from the frame. Electronic components are attached to the outer surface of each board. The components are mainly MSI and unit logic integrated circuits in flat-pack form (14- and 16-pin flat packs) and RAM and PROM devices in dual in-line packages. Some discrete components are used. Memory pages employ a folded, cast metal frame. Integrated circuit devices on memory pages include SSI and MSI TTL, linears and hybrids.

Heat generated by the components is conducted through the MIB to the metal page frame. Conduction through the frame and mounting flange thermal interface cools the page.

The MIB's are made of several layers of etched copper-clad, epoxy-glass laminates bonded together under heat and pressure. Connections between conductor layers are made through plated holes.

Integrated circuit flat packs are soldered to the etched patterns on the surface of the MIB's. Discrete components are soldered in plated holes or on the board surface, depending on the terminal configuration. A conformal coating is applied for environmental protection.

Two 98-pin connectors are fastened along the lower edge of the page frame. The contacts are a blade-and-fork configuration. Gaskets seal the contact interface and the space between the receptacle and the back panel.

Sixty-four feedthrough connections and 128 test points are located along the upper edge of the frame. The page is fastened to a supporting structure at the two mounting flanges (ears), with additional support provided at the connectors. Keyed guide pins project from the lower edge of the page to guide the page during insertion, to prevent mislocation, and to ensure that the correct page is installed in the proper location. Color coded dots on the LRU housing adjacent to the space replaceable unit (SRU) ear tiedown point and on the individual SRU pages assist in preventing mislocation. Jackscrews on the mounting flanges permit installation and removal without special tools.

A typical logic page is presented in figure 2-16. Figures 2-17 and 2-18 show several views of the CPU.

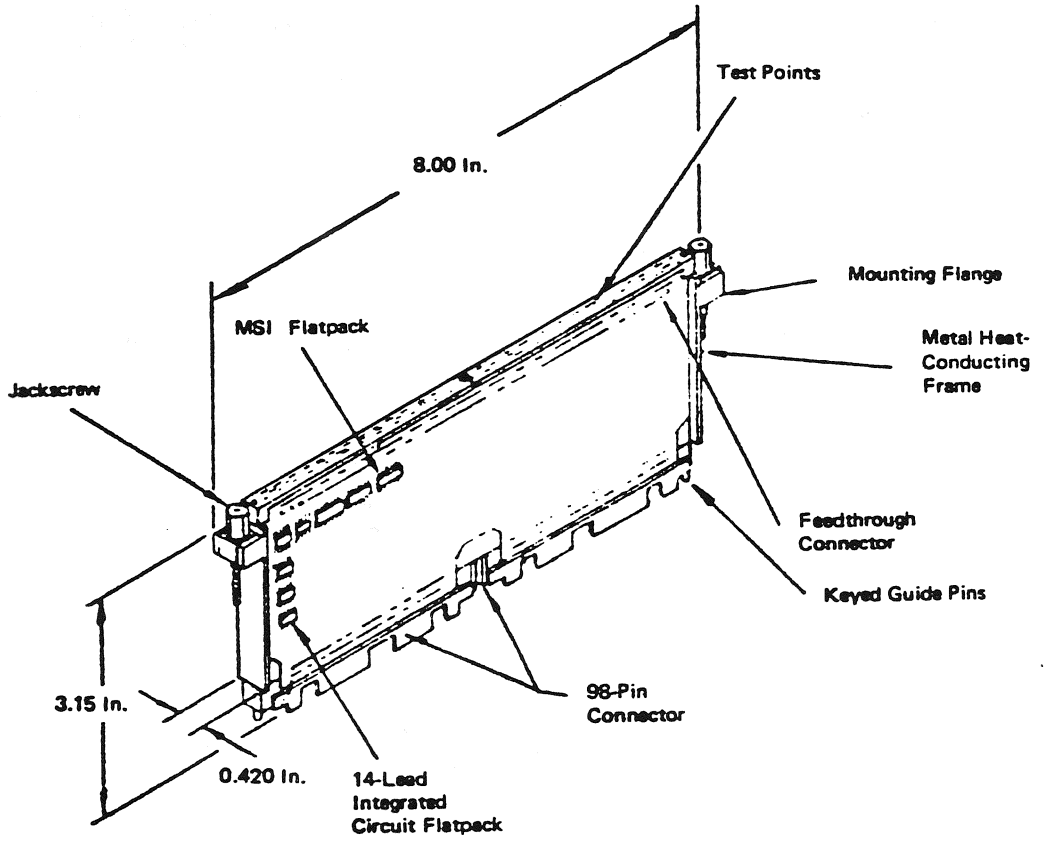


Figure 2-16.- Typical CPU and IOP logic page.

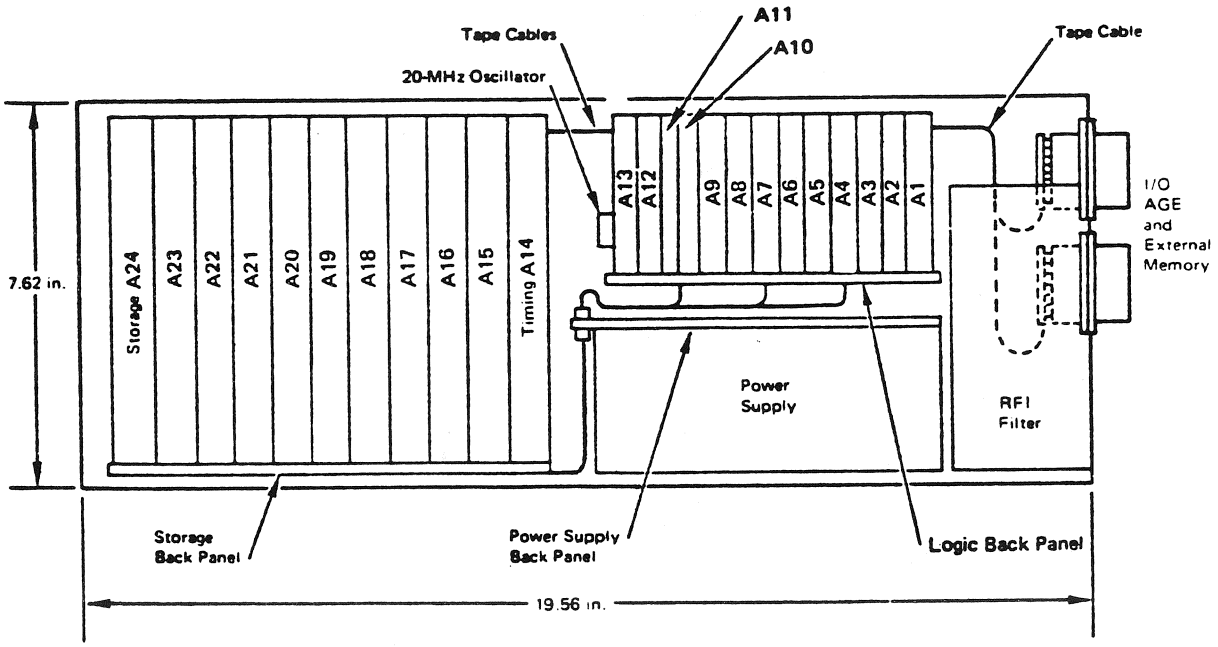


Figure 2-17.- Central processing unit (side view).

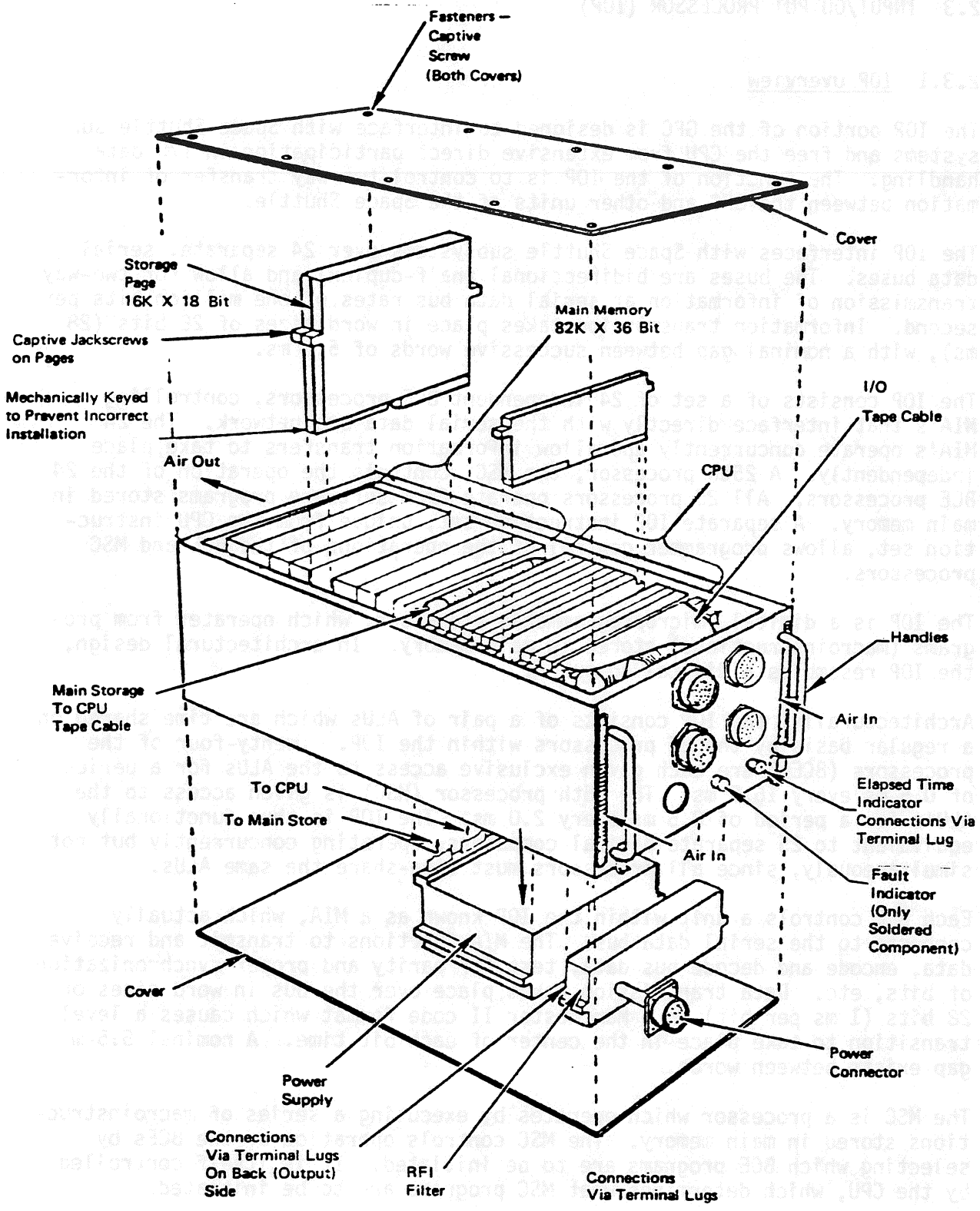


Figure 2-18.- CPU perspective illustration.

2.3 INPUT/OUTPUT PROCESSOR (IOP)

2.3.1 IOP Overview

The IOP portion of the GPC is designed to interface with Space Shuttle subsystems and free the CPU from extensive direct participation in I/O data handling. The function of the IOP is to control two-way transfer of information between the GPC and other units of the Space Shuttle.

The IOP interfaces with Space Shuttle subsystems over 24 separate, serial data buses. The buses are bidirectional (half-duplex) and allow for two-way transmission of information at serial data bus rates of one million bits per second. Information transmission takes place in word sizes of 28 bits (28 ms), with a nominal gap between successive words of 5.5 ms.

The IOP consists of a set of 24 independent BCE processors, controlling MIA's that interface directly with the serial data bus network. The 24 MIA's operate concurrently and allow information transfers to take place independently. A 25th processor, the MSC, controls the operation of the 24 BCE processors. All 25 processors operate from software programs stored in main memory. A separate IOP instruction set, unique from the CPU instruction set, allows programmer control of the operations of the BCE and MSC processors.

The IOP is a digital, microprogrammable processor, which operates from programs (macroinstructions) stored in main memory. In architectural design, the IOP resembles a digital computer.

Architecturally, the IOP consists of a pair of ALUs which are time shared on a regular basis by the 25 processors within the IOP. Twenty-four of the processors (BCEs) are each given exclusive access to the ALUs for a period of 0.5 ms every 16.5 ms. The 25th processor (MSC) is given access to the ALU's for a period of 0.5 ms every 2.0 ms. The IOP is thus functionally equivalent to 25 separate digital computers, operating concurrently but not simultaneously, since all processors must time-share the same ALUs.

Each BCE controls a unit within the IOP known as a MIA, which actually connects to the serial data bus. The MIA functions to transmit and receive data, encode and decode bus data, test for parity and proper synchronization of bits, etc. Data transmission takes place over the bus in word sizes of 28 bits (1 ms per bit) in a Manchester II code format which causes a level transition to take place in the center of each bit time. A nominal 5.5-ms gap exists between words.

The MSC is a processor which operates by executing a series of macroinstructions stored in main memory. The MSC controls operation of the BCEs by selecting which BCE programs are to be initiated. It is itself controlled by the CPU, which determines what MSC programs are to be initiated.

The IOP operates concurrently with the CPU; however, it basically functions under CPU control. The CPU controls the IOP by selecting the MSC programs to be executed, which in turn select the BCE programs. The CPU can disable or halt IOP operation at any time, either entirely (all BCEs and the MSC) or selectively (a single BCE). The CPU is the only device other than the Pulse Code Modulation (PCM) master which can enable MIAs for transmission and reception of serial bus data.

Within the GPC, both the IOP and CPU execute programs stored in main memory. Since the CPU and the 25 processors in the IOP act, in effect, as 26 digital computers operating 26 different programs and all accessing main memory for retrieval of program instructions, there is contention for access to main memory. This is resolved by two modes of operation. In one mode (normal interleave mode), the CPU and IOP interleave their memory requests, alternately sharing main memory on essentially an instruction-by-instruction basis. In the second mode (burst mode), the IOP is allowed exclusive control of memory access. During this mode of operation, the CPU is unable to access memory for execution of CPU instructions. Burst mode is entered whenever there are eight or more pending IOP memory requests. Normal mode operation takes place when there are fewer than eight IOP memory requests.

IOP memory requests from the 25 processors within the IOP take place through a memory request queue stack (64 requests) located in the IOP. Since main memory addressing logic is physically located in the CPU, IOP main memory requests take place across an 18-bit (functional) parallel address and 32-bit (functional) parallel data channel between the IOP and CPU. Main memory access can be performed at various rates depending on the memory access mode of operation. The rates are read burst (730K words (32 bits) per second), write burst (621K words per second), read normal (408K words per second), and write normal (373K words per second).

The IOP hardware elements are shown in figures 2-19 and 2-20. The elements consist of channel control, control monitor, processor elements, multiplexer input adapter, and Redundancy Management (RM) logic. These elements are described in the following paragraphs.

2.3.2 IOP Channel Control

The channel controller portion of the IOP provides the communication interface between the CPU and IOP. All signals between the IOP and CPU exist at this interface (except for those signals required to allow the 24K x 36 bits of core memory located in the IOP to function in conjunction with the 82K x 36 bits of memory located in the CPU). The signals between the IOP and CPU are of several basic types: a 35-bit bidirectional parallel data bus (32 bits of data, 2 parity bits, 1 storage protect bit); control signals associated with the 35-bit bidirectional parallel data bus; a 19-bit parallel address bus (IOP to CPU: 18 address lines, 1 parity line); 5 interrupt signals (IOP to CPU); and various timing, synchronization, and reset signals. A description of each signal and its function is provided in table 2-IV.

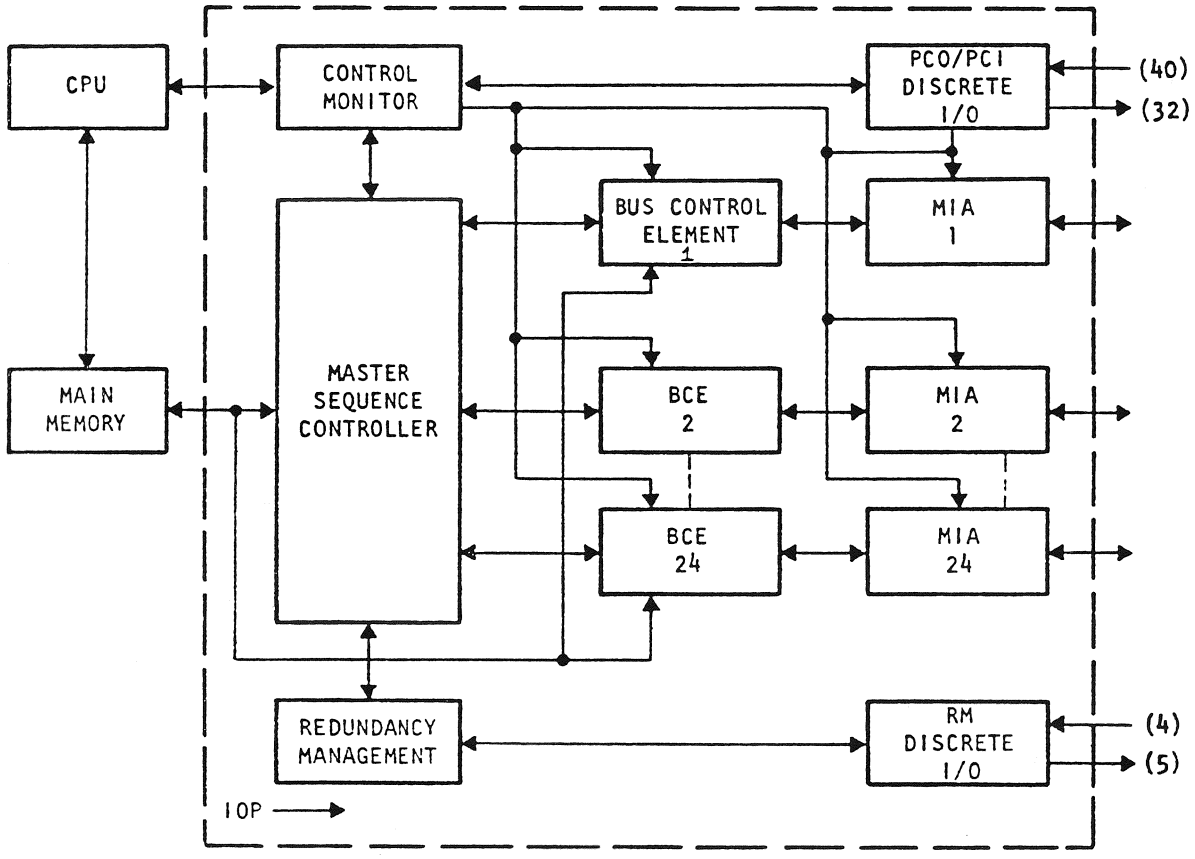


Figure 2-19.- Input/output processor interface.

The channel controller operates under two basic modes of operation; the modes are differentiated by which of the two units (IOP or CPU) initiates the information transfer. For CPU-initiated transfers, the 35-bit parallel data channel acts to transfer 32 bits of data between the CPU and IOP under the heading of direct I/O. The CPU-initiated data transfers are referred to as PCO and PCI. (The term program is used to identify the data transfer as taking place under CPU program control.)

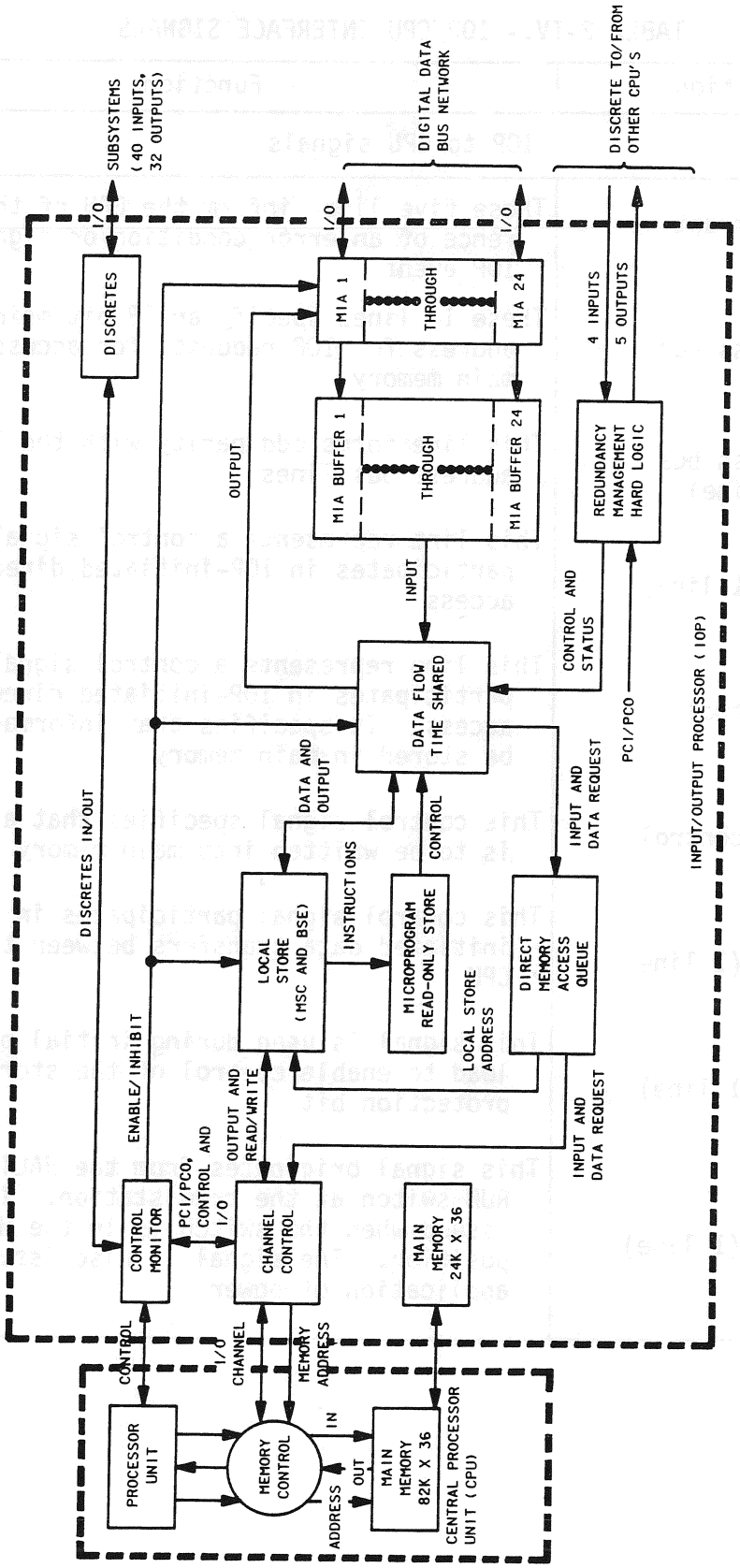


Figure 2-20.- Input/output processor.

TABLE 2-IV.- IOP/CPU INTERFACE SIGNALS

Description	Function
IOP to CPU signals	
External interrupt (5 lines)	These five lines inform the CPU of the occurrence of an error condition or significant IOP event
Channel address bus (18 lines)	These 18 lines specify an 18-bit main memory address for IOP requests for access to main memory
Channel address bus parity (1 line)	This line forms odd parity with the 18-channel address bus lines
DMA request (1 line)	This line represents a control signal which participates in IOP-initiated direct memory access
DMA store control (1 line)	This line represents a control signal which participates in IOP-initiated direct memory access. It specifies that information is to be stored in main memory
DMA halfword control (1 line)	This control signal specifies that a halfword is to be written into main memory
Data request (1 line)	This control signal participates in CPU-initiated data transfers between the IOP and CPU
IPL lockout (1 line)	This signal is used during initial program load to enable control of the storage protection bit
System reset (1 line)	This signal originates from the HALT/STANDBY/RUN switch at the crew station. It is issued when the switch is in the HALT position. The signal is also issued upon application of power

TABLE 2-IV.- Continued

Description	Function
CPU to IOP signals	
DMA priority (1 line)	This control signal allows a transfer to main memory for an IOP-initiated DMA request
Disable DMA (1 line)	This signal is issued by the CPU to disable (reset) the DMA channel during the power-down sequence
MS halfword parity (2 lines)	These bits indicate that a parity error was encountered during a main memory access for the DMA channel
Data acknowledge (1 line)	This control signal participates in both IOP- and CPU-initiated 32-bit transfers between the IOP and CPU
PCI/PCO control word gate (1 line)	This control signal informs the IOP that a PCI or PCO command is on the I/O channel
1 MHz clock (1 line)	This signal is used by the IOP channel control logic
Machine reset (1 line)	This signal is issued from the CPU power supply to indicate a power-off condition. It causes the IOP to be placed in the reset state
Interface reset (1 line)	This signal is used during error recovery to reset the IOP channel controller
Bidirectional signals (IOP to CPU; CPU to IOP)	
Data bus (32 lines)	This bus is the 32-bit parallel channel between the IOP and CPU

TABLE 2-IV.- Concluded

Description	Function
Data bus halfword parity (2 lines)	These lines specify halfword parity for the data bus for both direct I/O and DMA modes of operation
DMA storage protect or violation	This line either assigns storage-protect to the 32 bits being placed in main memory or indicates a storage-protect violation

Two consecutive 32-bit transfers normally take place for each PCO- or PCI-initiated transfer. For PCO transfers, the two 32-bit transfers take place from CPU to IOP. The first 32-bit word is in the form of a command to the IOP, and the second 32-bit word contains data which are to be transferred to the IOP.

For PCI transfers, the first 32-bit transfer contains a command to the IOP to transmit 32 bits of data back to the CPU. The second 32-bit transfer then takes place from the IOP to CPU to complete the PCI word transmission pair.

To prevent a failure in the IOP from holding the CPU in a state awaiting an IOP response to the PCO or PCI transfer, the CPU performs several timeout checks. These timeout checks terminate the channel transfer if the IOP does not respond to the first word within 8.4 microseconds and with (or to) the second word within 6.0 (or 8.4) microseconds.

The second mode of operation associated with the IOP channel controller is for information transfers which are initiated by the IOP. In this mode of operation, the 35-bit bidirectional channel acts under the heading of a DMA channel. The IOP addresses the main memory addressing logic in the CPU via the 18 address lines and either places or extracts 32 bits of information from main memory via the 35-bit bidirectional data lines. In this mode of operation, the IOP acts on a CPU main memory cycle-steal basis; that is, it accesses main memory during CPU instruction execution by stealing access to main memory when memory is not being used by the CPU. Cycle-stealing has some effect on CPU performance. The cycle-stealing action can cause the CPU to wait for access to main memory for a maximum period of approximately 1.7 microseconds for each memory access.

The IOP-initiated requests for access to main memory are issued from a DMA queue stack. The requests may come about for any of three reasons: IOP requests for the next MSC or BCE macroinstructions for program execution, IOP requests for data which are to be transmitted to Space Shuttle subsystems, and IOP requests for main memory access to place data received from

Space Shuttle subsystems into main memory. The DMA queue is capable of holding 64 requests. An attempt to place more than 64 requests in queue will result in an interrupt.

A single request from the queue requires an average main memory access time of approximately 2.5 microseconds. When more than seven requests have been stacked, the DMA enters a burst mode of operation, which has an average main memory access time of 1.4 microseconds. During burst mode operation, CPU program execution is stopped and the DMA channel has exclusive access to main memory. When the queue is bled down to less than eight requests, burst mode operation ceases.

The IOP channel controller performs the following functions.

A. A 32-bit input (to CPU) data register holds 32 bits of information which is to be transferred to main memory. The contents of the register may be the result of direct I/O (PCI) or DMA utilization of the channel.

B. A 32-bit output (from CPU) data register holds 32 bits of information which is being transferred from main memory to the IOP. The contents of the register may be the result of direct I/O (PCO data) or DMA utilization of the channel.

C. An 18-bit PCI/PCO command register holds the 18-bit PCI/PCO command contained within the first 32-bit word of the PCI/PCO two-word transfer.

D. An 18-bit DMA address register holds the 18-bit main memory address for DMA utilization.

E. In addition to the registers, the channel controller also contains parity generating and checking circuitry. Parity is associated with the 35-bit bidirectional data channel bits (1 parity bit per halfword) and the 19-bit address channel (18 address bits, 1 parity bit).

In summary, the channel control logic performs the control functions associated with the direct I/O and DMA modes of operation. It establishes the necessary links between the various IOP elements and the DMA channel. It controls PCI/PCO and the DMA activity.

2.3.3 IOP Control Monitor

The control monitor provides many of the miscellaneous control functions internal to the IOP. These include control of power-on/off sequencing and resets, control of discrete outputs, and control of MIA transmitters and receivers. The control monitor can also place the MSC or any of the BCE processors in the halt state, thereby halting processor operation. The control monitor functions under CPU control via PCI and PCO.

The control monitor has the capability to accept the following from the CPU via a PCO command: 24 MIA transmitter-enable signals; 24 MIA receiver-enable signals; 32 discrete output signals (for transmission to subsystems external to the IOP); and 25 processor halt signals (MSC and 24 BCEs).

To assist CPU determination of the present IOP status, the above signals are stored in the IOP and may be read by the CPU via a PCI command. The signals, however, are not capable of being changed within the IOP by any action of the IOP processors.

Of the 32 discrete output signals, only 31 are under actual PCO control. The remaining discrete output signal is wired in the IOP to a fixed logic one state. It is issued from the IOP and routed through an external connector back to 3 of the 40 IOP discrete inputs. The discrete output signal forms the source signal for GPC self-identification. The three input signals form a code that establishes the identify of the GPC within the Space Shuttle Orbiter.

In addition to the above, the control monitor allows the CPU to monitor several other categories of signals via a PCI command. These consist of: 40 discrete inputs (from subsystems external to the IOP); 32 RM status signals; and a total of 32 interrupt status discretes, which may be read in groups of 6, 6, 12, 4, and 4 bits, respectively, by appropriate PCI commands. The five groups of interrupt status discretes are associated with the five interrupt signals transmitted from the IOP to the CPU. The CPU services one of the IOP-generated interrupts, in part, by reading the associated set of interrupt status discretes, via a PCI command, to determine the cause of the IOP interrupt.

To support control monitor operation, the following hardware elements are provided: a 24-bit MIA transmitter-enable register (1 bit per MIA); a 24-bit MIA receiver-enable register (1 bit per MIA); a 32-bit discrete output register, a 25-bit halt processor register (1 bit per processor); and 5 interrupt status registers consisting of 6, 6, 12, 4, and 4 bits, respectively.

The 32-bit RM status register is contained in the RM logic section of the IOP. The 40 discrete inputs do not have an associated register, since the signals are generated external to the IOP and are provided to the IOP as inputs.

All of the above signals are gated to, or received from, the channel controller by multiplexers contained within the control monitor. One multiplexer is provided for PCO. Two multiplexers are provided for PCIs; these multiplexers are labeled control monitor PCI and RM PCI.

2.3.4 IOP Processor Elements

IOP processing is accomplished by MSC and BCE software programs which are located in main memory. The MSC and BCE program instructions are brought to the IOP one at a time for execution. Each instruction may be thought of as a macroinstruction since it causes a sequence of events to take place in the IOP; i.e., each macroinstruction is decoded and is executed as a sequence of microinstructions. There are 45 basic MSC macroinstructions which are implemented by 262 microinstructions, for an average of approximately 6 microinstructions for each macroinstruction. The basic BCE macroinstructions total 22 and are implemented by 261 microinstructions.

The MSC and BCE processors may be regarded as consisting of shared data flow and dedicated local store elements. The data flow element consists of a pair of ALUs with six associated working registers. The data flow element is time shared by all BCE and MSC processors. Local store consists of sets of 16 registers permanently assigned to each processor. The registers are arranged in three banks, A, B, and C, with 4, 4, and 8 registers, respectively. All registers are 18 bits. The processors are sequenced and controlled by a master timer, which allocates the ALUs for individual MSC or BCE use. In addition, the processors have access to a unit capable of decoding the individual macroinstructions into a set of microinstructions. This unit is known as the microstore and decode unit. It also holds the address of the next microinstruction which is to be executed for each BCE.

Figure 2-21 illustrates the basic processing elements. Each processor (MSC or BSE) is permanently allocated 16 words of local store (18 bits per word) arranged in 3 banks of 4, 4, and 8 words. The 16 words act, in effect, as 16 registers (18 bits per register).

The master timer determines which processor is to be active; i.e., which processor is to be allocated a microcycle of 0.5 microsecond for processing. During this microcycle, the processor has access to the ALUs. The master timer allocates a 0.5-microsecond microcycle to each of the 24 BCE processors once every 16.5 microseconds. The MSC is allocated a 0.5-microsecond microcycle once every 2.0 microseconds. This higher duty cycle is necessary since the MSC basically controls and monitors the individual BCE operations, along with occasional CPU intervention, via PCI/PCO commands.

When a processor is allocated a microcycle, the address of the microinstruction contained in microstore and the contents of local store for that processor are used to determine the processing which is to take place. The local store contents for the MSC and BCE processors are listed in table 2-V.

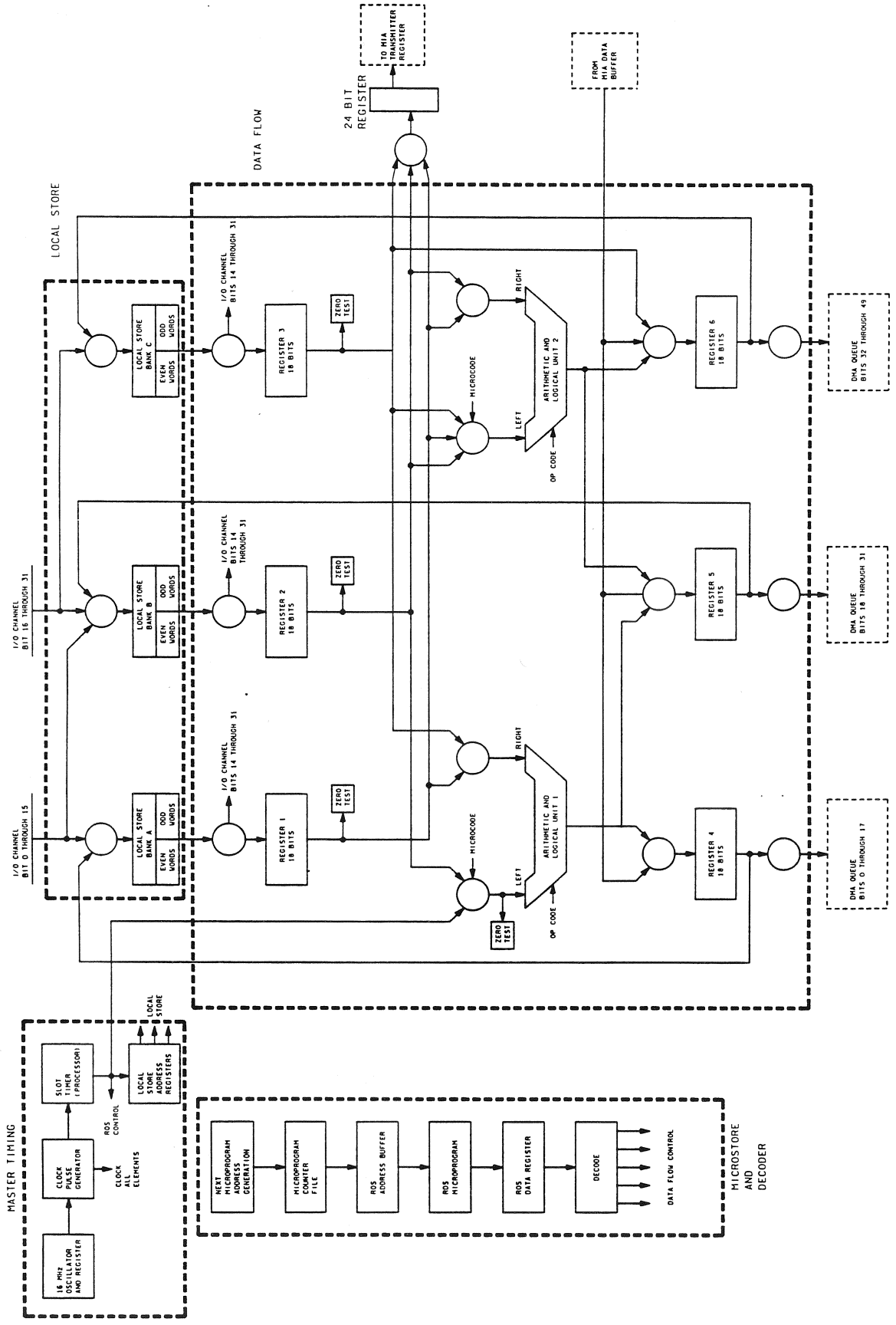


Figure 2-21.- Processor elements data paths.

TABLE 2-V.- LOCAL STORE MAPS

Bank A	Bank B	Bank C
MSC local store map		
0 (Data address) 1 2 Program counter 3 Index register	0 (Data high) 1 2 Instruction high 3 Accumulator high	0 (Data low) 1 2 Instruction low 3 Accumulator low 4 5 6 External call register 7 MSC status
BCE local store map (one of 24 maps)		
0 (Buffer address) 1 (Transfer count) 2 Program counter 3 (Index base)	0 (Data high) 1 2 (Instruction high) 3 Max. time out	0 (Data low) 1 2 (Instruction low) 3 Base register 4 2 x BCE No. (identifier) 5 Interface unit address 6 BCE status high 7 BCE status low

The registers in the three local store banks that are noted are those registers (shown without parentheses above) which are programmer visible via instructions of the MSC and BCE instruction sets. The remaining registers of local store are used as working registers by the microprogram when executing the macroinstruction for that processor.

The programmer visible registers in the local store banks are as follows.

A. The 18-bit program counter indicates the main memory location of the macroinstruction presently being executed.

B. The 18-bit index register and the 18-bit base register are used for generating 18-bit main memory addresses for MSC and BCE program execution.

C. The instruction high is an 18-bit register which contains the high order 16 bits of the present MSC instruction. The 16 bits are right justified; i.e., they are located in bits 2 through 17 (bit positions 0 and 1 are ignored).

D. The instruction low is an 18-bit register which contains the low order 18 bits of the last MSC instruction.

E. The 32-bit MSC accumulator is held within two 18-bit registers of MSC local store. The accumulator is capable of accumulating a full word of data from main memory. MSC instructions are available to load, modify, test, and store the 32-bit accumulator. Accumulator high contains the high order 16 bits of the 32-bit accumulator, right justified (bit positions 0 and 1 are ignored). Accumulator low contains the low order 16 bits of the accumulator, also right justified.

F. The 18-bit MSC status register holds the status of the MSC processor.

G. The 18-bit external call register is loaded by CPU PCO and contains the starting address of a program that is to be executed by the MSC. The MSC detects the presence of the external call register program by executing a sample external call instruction. Upon accepting the external call register program, the MSC sets the register to a value of all zeros.

H. The 32-bit BCE status register holds a 32-bit quantity in two 18-bit registers of local store. The 32 bits indicate the status of a BCE processor. Each 18-bit register holds 16 bits, right justified.

I. The 18-bit maximum timeout register is used primarily during the reception of data from a MIA to indicate the maximum time a BCE should wait for a subsystem to respond with the first word of data. This time is defined as the latency of the subsystem. The LSB of the register is 16.5 ms.

J. The 5-bit interface unit address is contained in an 18-bit local store register. The five bits represent the address of the subsystem presently in communication with the BCE.

The remaining registers of local store are used as temporary working registers during execution of the present macroinstruction. The registers are under microprogram control. At times, certain of the registers act as buffers for data which are being transmitted and received by the MIA of a particular BCE.

During a single microcycle time period of 0.5 ms, one microinstruction is executed. The microinstruction normally makes use of the pair of ALU's and six associated working registers.

The data flow element includes six working registers designated R1 through R6, as illustrated in figure 2-21. Registers R1 through R3 act as input buffers to the ALUs and are loaded from local store. Registers R4 through R6 act as output buffers for the ALUs and are capable of transferring information back into local store.

Registers R1 through R6 also take part in the transmission and reception of data from the MIAs. As illustrated in figure 2-21, registers R1 through

R3 are capable of transmitting data to the MIAs, registers R4 through R6 are capable of accepting MIA data.

The operation of the master timing and microprogram is described as follows. Each 0.5-microsecond microcycle consists of eight 62.5-nanosecond wide clock pulses. These clock pulses are derived from a 16-MHz crystal controlled square wave generator which is counted down by an eight-bit shift register arranged as a ring counter. During each clock pulse, one or more transfers of information or processing steps take place. The entire set of eight clock pulses is used to complete a microinstruction. Depending upon the microinstruction, the first two clock pulses of the next microcycle may be used to complete microinstruction address generation. Use of these two clock pulses does not interfere with the processor normally assigned to the new microcycle.

During a given microcycle, operation begins by determining which microinstruction is to be executed. This is accomplished by reading the contents of a microprogram counter file, which contains the next microprogram address for each processor. The microprogram address is then used to fetch the actual microprogram instruction from a 1.5K x 72-bit read-only memory. Each microprogram instruction is defined by 72 bits.

2.3.5 IOP MIA

There are 24 MIAs within the IOP which directly interface with the 24 buses of the DPS serial data bus network. The 24 MIAs are controlled by the 24 BCE's. The MIAs convert 24 bits of parallel IOP data into serial data, add a sync pattern and a parity bit, and encode the data for the bus. Conversely, the MIAs also receive data from the bus; decode it; check the sync pattern, number of bits, and parity; and convert it into parallel data for the IOP. The MIAs function asynchronously with respect to the controlling BCE's. A one-word MIA buffer is provided to allow MIA to BCE transfers of information for data which are received by the MIAs. The one-word buffer is not used for data transmissions from the MIA to other subsystems of the Space Shuttle.

2.3.6 IOP Redundancy Management Logic

The RM logic consists of fail vote logic and a watchdog timer.

During GPC operations, the CPU may determine that the GPC is self-failed, or in the PASS, the CPU may determine that another member of the common or redundant set is failed. One way in which these failures are annunciated is through the IOP fail vote logic.

The fail vote logic has five discrete outputs. One of these discretes is the GPC self-fail vote. The other four discretes are fail votes against the remaining four GPCs (one fail vote discrete runs to each of the other GPC's). The GPC fail vote outputs are controlled from the CPU via a PCO command. The status of all five output fail discretes from each GPC is

shown on a five by five matrix of lights on panel O1 called both the Computer Annunciator Matrix (CAM) and the Computer Status Matrix (CSM) (fig. 2-22).

The fail vote logic has four discrete inputs, one fail vote from each of the other four GPCs. If a GPC receives two fail votes from other GPCs, the fail vote logic will inhibit the output of fail votes to other GPCs and will issue the GPC's self-fail discrete to the CAM. It should be noted that GPC fail votes are only used as an indication of a GPC problem. Fail votes against a GPC cannot by themselves cause the GPC to fail to sync.

The watchdog timer is a 12-bit incrementing clock with a resolution of 0.768 ms and a total capacity of 3.145728 seconds. A correctly cycling CPU will periodically reload the timer prior to its timing out via a PCO command. If a watchdog timer timeout occurs, a timeout latch is set in the IOP, and the self-fail discrete is issued, which lights the failed GPC's I-FAIL CAM light. The watchdog timer is used only by the BFS and IPL software; it is not activated by the PASS.

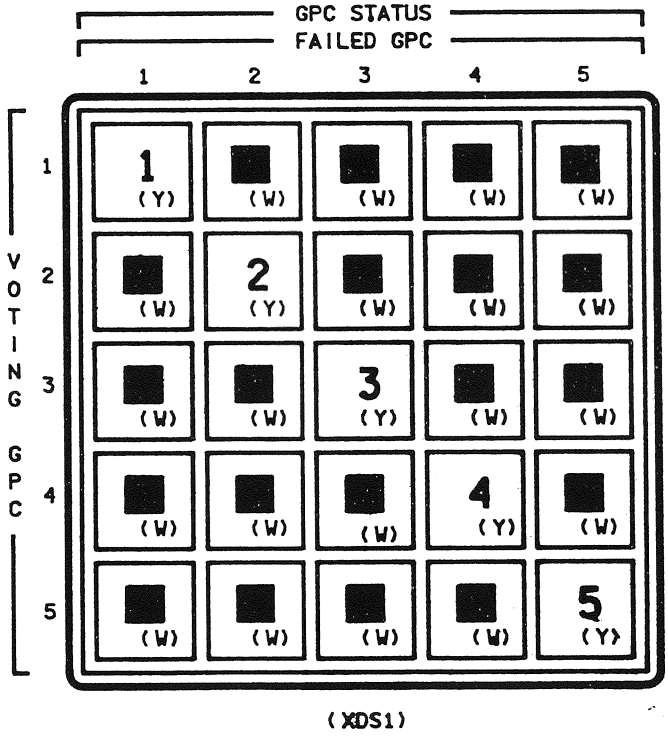


Figure 2-22.- Computer annunciation matrix (panel O1).

2.4 GPC DISCRETE INPUT/OUTPUT

The discrete inputs and outputs of the GPC are used for communicating discrete events between outside elements and the GPC. Table 2-VI is a list of the discrete inputs/outputs and a brief description of the function of each one.

2.4.1 Discrete Input/Output Circuits

Although discretes represent a single signal, they are transferred external to the GPC via two lines using a true/complement logic. True/complement logic converts a single on/off signal to two lines of opposite states. The ON state of a switch would provide an electrical ground (low) to the true line. At the same time, an open circuit would be provided at the switch on the complement line. Because of the type of logic used in the IOP receivers, an open circuit on a line results in the line's being at an electrical high. At the IOP receiver, the true and complement lines are converted back to a single signal with a value of "1", or "0". Figure 2-23 shows some typical examples of discrete inputs and outputs. Table 2-VII shows the truth table for discrete inputs.

The discrete output circuits, which also use true/complement logic, are used to drive talkbacks, lights, or discrete input circuits. The outputs provide a ground (true condition) or an open circuit (complement) to the attached device.

This is accomplished by using a transistor circuit called "open-collector" outputs. With this type of output, the active condition provides a ground and the complement provides an open circuit.

TABLE 2-VI.- DESCRIPTION OF DISCRETE INPUTS/OUTPUTS

(a) Discrete inputs

DI No.	Discrete function	Description
0	Halt command	From GPC MODE sw (pnl O6). The HALT input is routed through the BFC prior to going to the GPC.
1	Standby command	
2	Run command	
3	IPL command	From IPL pushbutton (pnl O6)
4	MMU 1 IPL command	From IPL source sw (pnl O6)
5	MMU 2 IPL command	
6	MMU 1 ready	Status discrete from MMU 1 (2)
7	MMU 2 ready	
8	GPC W BFS run	This discrete is used by the BFS to indicate to the PASS when it is in RUN. When the BFS is in RUN, the PASS will relinquish control of the payload buses, and the BFS will pick up control.
9	GPC X BFS run	
10	GPC Y BFS run	
11	GPC Z BFS run	
12	Spare	
13	I/O TERM B	This discrete comes from the BFC. Its state is determined from the GPC OUTPUT sw (pnl O6) and the status of the BFS engage discretes at the BFC. When the I/O TERM B is set, the IOP is inhibited via hardware from making outputs on the flight critical buses.
14	Spare	
15	Hardware memory dump	From GPC MEMORY DUMP sw (pnl MO4F2F). This switch is used to select a GPC for HISAM dump.
20	GPC W sync 1	These discrete lines are used for PASS GPC synchronization. A set of three lines (sync 1, sync 2, and sync 3) are output from each GPC to the other four GPC's. Also see DO No. 20, 24, and 28.
21	GPC X sync 1	
22	GPC Y sync 1	
23	GPC Z sync 1	
24	GPC W sync 2	
25	GPC X sync 2	
26	GPC Y sync 2	
27	GPC Z sync 2	
28	GPC W sync 3	
29	GPC X sync 3	
30	GPC Y sync 3	
31	GPC Z sync 3	
32	Station ID bit 0	The cable harness is uniquely wired for each IOP slot 1 through 5 to set these discrete inputs from the station ID output. These discretes are read during GPC IPL to allow the GPC to determine which GPC (GPC 1-5) it is. Since these discretes read during IPL, a failure of one of these discretes after IPL will have no impact.
33	Station ID bit 1	
34	Station ID bit 2	
35	BFS engage 1	From BFC module. Used to determine if BFS is engaged.
36	BFS engage 2	
37	BFS engage 3	

TABLE 2-VI.- Continued

DI No.	Discrete function	Description
38 39	CRT select A CRT select B	These discretes come from the BFC CRT SEL switch (pnl C3) and are routed through the BFC to the GPC. This switch is one means of changing CRT control between PASS and BFS.

(b) Discrete inputs (hardware RM)

DRI No.	Discrete function	Description
03 04 05 06	GPC W fail vote GPC X fail vote GPC Y fail vote GPC Z fail vote	Fail vote discretes from other GPCs. (See IOP Redundancy Management for more information.)

(c) Discrete outputs

DO No.	Discrete function	Description
07	I/O active indicator	Drives GPC OUTPUT talkback (tb) (pnl 06). Talkback has two positions, barberpole and gray.
09	Run indicator	Drives GPC mode tb (pnl 06) to RUN position. Talkback has three positions - RUN, barberpole, and IPL.
20	Sync 1	Used in PASS GPC synchronization. See DI No. 20-31.
22	BFS run	Used by BFS GPC to indicate to PASS that BFS is in RUN (see DI No. 8-11 description).
24 28	Sync 2 Sync 3	Used in PASS GPC synchronization. Also see DI No. 20-31.
30	ID source	Used in determining GPC ID (see DI No. 32-34 description).
31	IPL indicator	Drives GPC MODE tb (pnl 06) to IPL. Talkback has three positions - RUN, bp, and IPL.

(d) Discrete outputs (hardware RM)

DRO No.	Discrete function	Description
00 07 08 09 10	GPC V fail indicator GPC W fail indicator GPC X fail indicator GPC Y fail indicator GPC Z fail indicator	GPC vote discrete outputs. (See IOP Redundancy Management.)

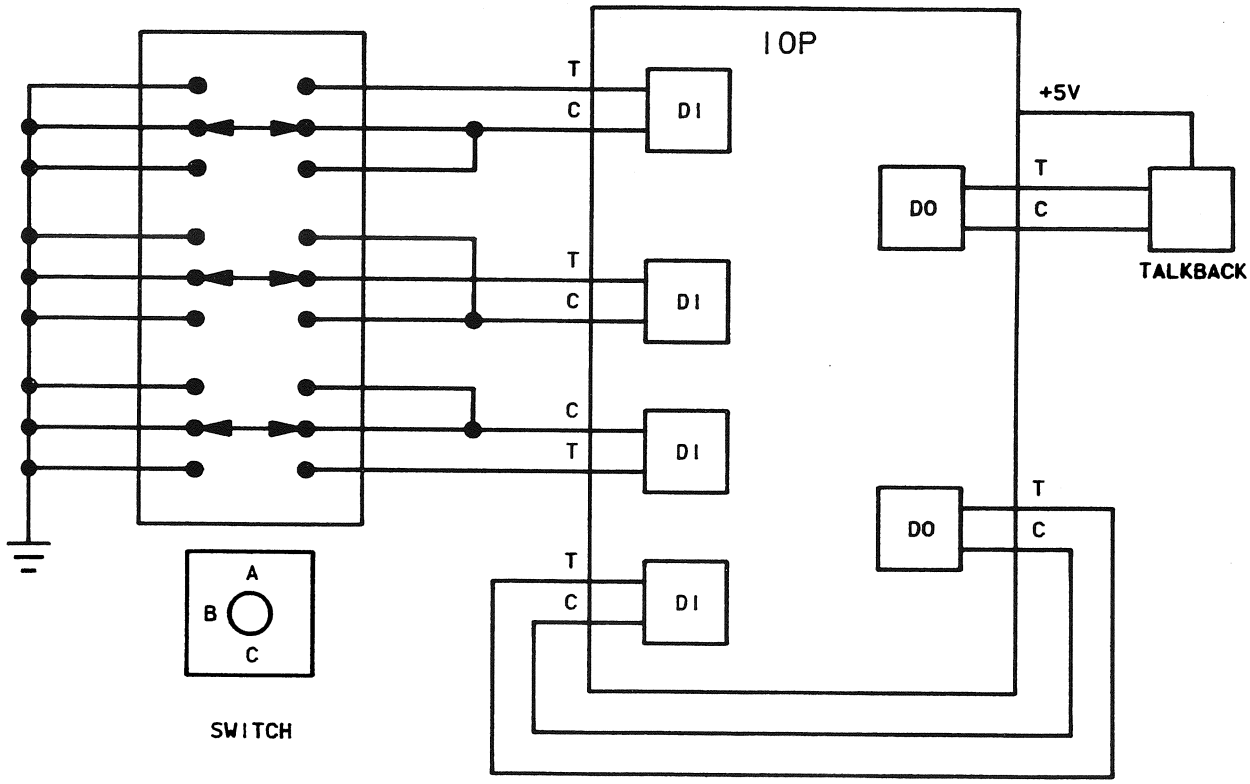


Figure 2-23.- Typical input/output usage.

TABLE 2-VII.- DISCRETE INPUT RECEIVER TRUTH TABLE

Input true line	Input complement line	Logical output	Comments
High	High	0	Failure condition
High	Low	0	Complement present
Low	High	1	True present
Low	Low	Undefined	Failure condition. Logical output cannot be predicted.

2.4.2 GPC IOP Input Discrete Hardware Commonality

The GPC receives discrete inputs via hybrid differential receivers located in the IOP. Each of these receiver units receives both the true and the complement lines for four different discretes. If it is discovered that any discrete input to a single GPC is failed in a high or low state, then the other three discretes which share the failed discrete's receiver must be considered suspect. Table 2-VIII shows the groupings for the discrete inputs.

TABLE 2-VIII.- DISCRETE INPUTS

Receiver	DI No.	Discrete function
1	0	Halt CMD
	1	Standby CMD
	2	Run CMD
	3	IPL CMD
2	4	MMU 1 IPL CMD
	5	MMU 2 IPL CMD
	6	MMU 1 ready
	7	MMU 2 ready
3	8	GPC W BFS run
	9	GPC X BFS run
	10	GPC Y BFS run
	11	GPC Z BFS run
4	12	Spare
	13	I/O term B
	14	Spare
	15	Hardware memory dump
5	16	Spare
	17	Spare
	18	Spare
	19	Spare
6	20	GPC W sync 1
	21	GPC X sync 1
	22	GPC Y sync 1
	23	GPC Z sync 1
7	24	GPC W sync 2
	25	GPC X sync 2
	26	GPC Y sync 2
	27	GPC Z sync 2
8	28	GPC W sync 3
	29	GPC Y sync 3
	30	GPC Y sync 3
	31	GPC Z sync 3
9	32	Station ID bit
	33	Station ID bit
	34	Station ID bit
	35	BFS engage 1
10	36	BFS engage 2
	37	BFS engage 3
	38	CRT SEL A
	39	CRT SEL B

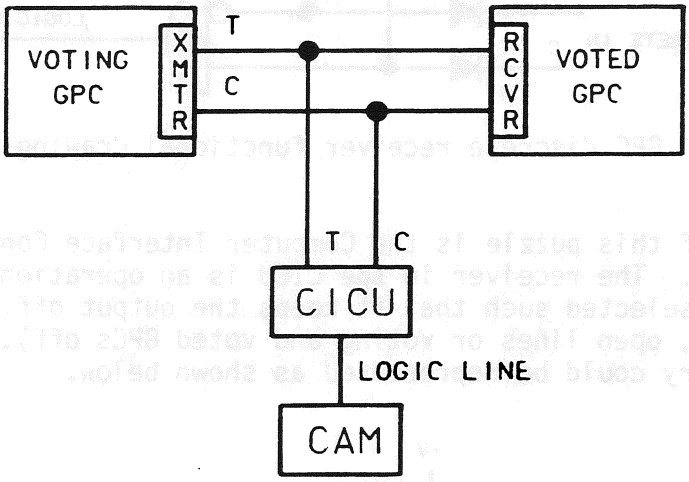
Discrete inputs (Hardware RM)

Receiver	DRI No.	Discrete function
1	03	GPC W fail vote
	04	GPC X fail vote
	05	GPC Y fail vote
	06	GPC Z fail vote

2.4.3 GPC Fail Vote Discretes/CAM Interface

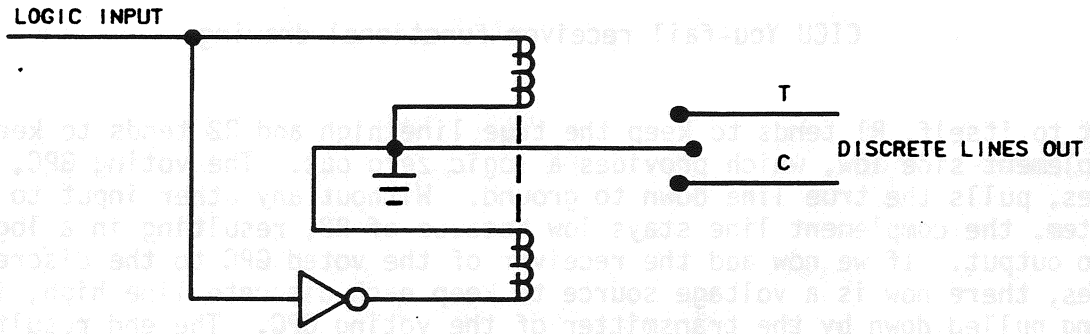
The discrete circuitry involved in driving the CAM matrix GPC fail lights has some unique designed-in features which warrant discussion. Two cases will be covered here, the GPC I-fails (vote against self) and the GPC You-fails (vote against another GPC).

For the You-fail cases, there are three terminations on the discrete data lines; the voting GPC, the voted-against GPC, and the CICU, which drives the CAM matrix (see below).



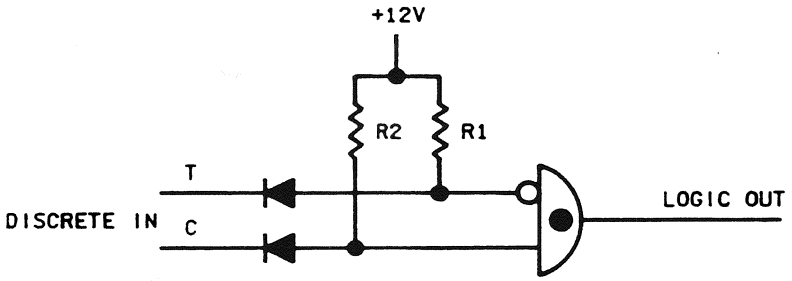
You-fail discrete system

If the voting GPC is on, either the true (T) or complement (C) line will be grounded, and the opposite line left floating. When the GPC is powered off, neither line is grounded and both are left floating. Therefore, the transmitters have pulldown capability only. A functional representation of the transmitter is shown below.



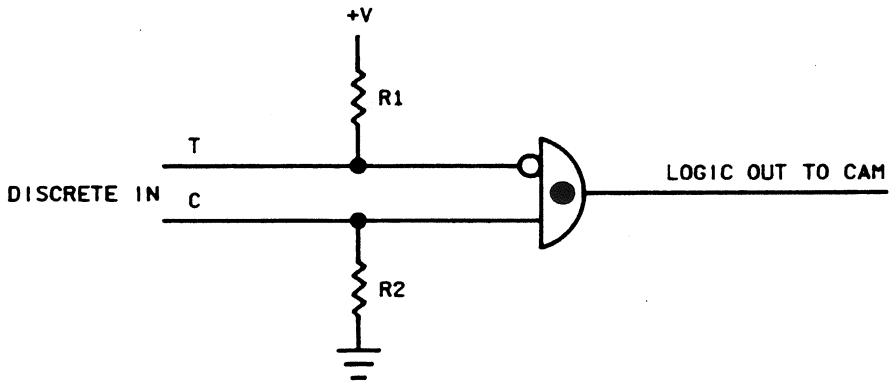
Discrete transmitter functional drawing

The receivers perform two functions. Not only do they sense the T and C lines to determine logic status, they also provide drivers to supply a pullup voltage to the discrete line left floating by the transmitter. A functional representation of the receiver is shown below.



GPC discrete receiver functional drawing

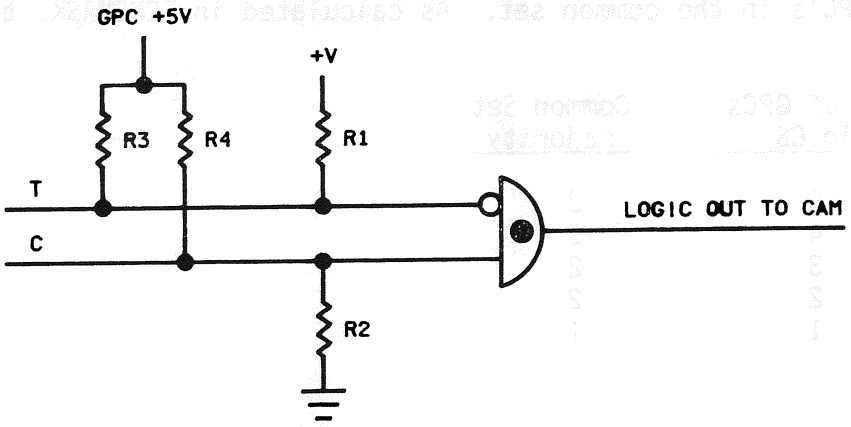
The third part of this puzzle is the Computer Interface Conditioning Unit (CICU) circuitry. The receiver in the CICU is an operational amp with external biases selected such that it keeps the output off, given a lack of any inputs (i.e., open lines or voting and voted GPCs off). The CICU receiver circuitry could be represented as shown below.



CICU You-fail receiver functional drawing

Left to itself, R1 tends to keep the true line high and R2 tends to keep the complement side low, which provides a logic zero out. The voting GPC, as it votes, pulls the true line down to ground. Without any other input to the system, the complement line stays low because of R2, resulting in a logic zero output. If we now add the receiver of the voted GPC to the discrete lines, there now is a voltage source to keep each discrete line high, if not being pulled down by the transmitter of the voting GPC. The end result is that, if a voted-against GPC is turned off, the CICU can no longer recognize a fail vote, and the CAM light will go out.

The second case is that of the I-fail discretes, the GPC discrete transmitter is the same, but the CICU receiver circuitry is different. Instead of having another GPC's discrete receiver on the line, the C&W system shares the lines. However, C&W plays no active role on the self-fail lines, it merely monitors them. The CICU I-fail receiver could be represented functionally as shown below.



CICU I-fail receiver functional

The resistors R1 and R2 perform the same function as before, that is keeping the logic output low, given no inputs. The GPC +5 V power, and resistors R3 and R4 act to provide the extra pullup on the lines.

These discussions show why the CICU cannot recognize fail votes and light CAM lights without external help. This explains why powering off a GPC will also turn off that GPC's column of CAM lights.

2.4.4 PASS GPC Discrete Input RM

Table 2-IX shows the discretes that are Redundancy Managed (RM'd) in the PASS (by GPC-Discrete Redundancy-Manager, FCMDSCRM). RM means that rather than each GPC using its own raw discrete input, it uses the discrete input value as determined by a majority of common set members.

The Common-Set-Majority term (TCVTMGCS) is calculated as greater than one-half of the GPC's in the common set. As calculated in FCMSMASK, the values are

No. of GPCs in CS	Common Set Majority
5	3
4	3
3	2
2	2
1	1

Each time the discretes are to be RM'd, the number of GPCs in the common set is checked. If there is only one GPC in the common set, then the raw discretes become the RM'd discretes. If the common set contains two or more GPCs, then a count is taken of the number of GPC's detecting a change in a particular raw discrete. If the count of GPCs detecting a change is greater than or equal to the Common Set Majority calculated for this case, then the changed discrete is entered as the RM'd discrete.

If you have a common set of two GPCs, the majority is defined as two. If one GPC sees MM1 go ready, but the other GPC sees it as still busy, then the RM'd discrete would stay busy and MM1 would not be available to either GPC.

TABLE 2-IX.- REDUNDANCY MANAGED DISCRETES

DI No.	Discrete function	Comments
4	MMU 1 IPL command	Determine position of IPL source sw.
5	MMU 2 IPL command	
6	MMU 1 ready	Determine MMU busy/ready status.
7	MMU 2 ready	
8	GPC W BFS run	Determine ID of BFS in RUN (if any).
9	GPC X BFS run	
10	GPC Y BFS run	
11	GPC Z BFS run	
20-31	GPC sync discretes	These sync discretes are RM'd for "000" pattern only.
35	BFS engage 1	These discretes are RM'd for disengage only.
36	BFS engage 2	
37	BFS engage 3	
38	CRT SEL A	Determine position of BFC CRT SEL sw.
39	CRT SEL B	

2.4.5 Failure of GPC Mode Discrete Inputs

2.4.5.1 GPC Mode Discrete Circuitry

To explain the IOP hardware response to failure conditions, the normal response of the hardware will first be discussed. The mode switch discretes (HALT, STBY, and RUN), after passing through the discrete receivers of the IOP, enter a special latching circuit (fig. 2-24). In the discrete receiver circuit, the true and complement lines from the mode switch are converted into a single signal. This signal is applied to the preset input of a D-type flip-flop (F-F). The preset input causes the Q output to a logical 1 state. Both the preset and clear inputs perform their respective functions when the input is at a logical 0.

A transition of any discrete (HALT, STBY, or RUN) from the false state to the true state will cause the corresponding F-F to be reset. The transition of any discrete from true to false will have no effect on the circuit. This implies the F-Fs remain in their last valid state when any or all inputs are removed.

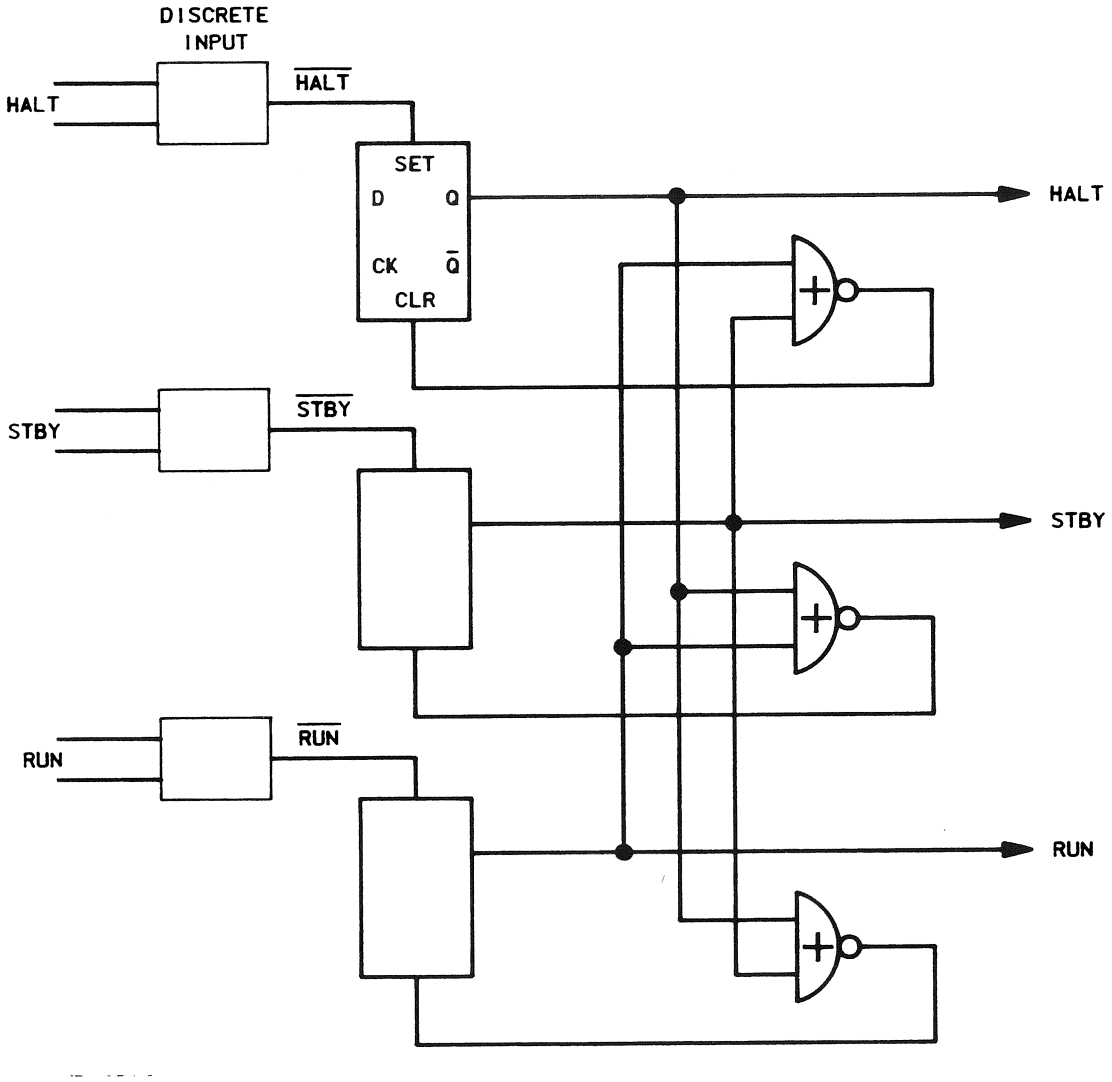


Figure 2-24.- Input circuit.

With three inputs there are eight different configurations the inputs may assume. Table 2-X shows the possible combinations and the hardware response to each combination.

TABLE 2-X.- INPUTS VS. HARDWARE RESPONSE

Inputs			Hardware response
Halt	Stby	Run	
0	0	0	Last valid state ①
0	0	1	Norml RUN state
0	1	0	Normal STBY state
0	1	1	Software response only
1	0	0	Normal HALT state
1	0	1	HALT
1	1	0	HALT
1	1	1	HLT

①If the GPC is power cycled, the last state cannot be guaranteed.

2.4.5.2 HALT Discrete Failure

From Table 5-XV it can be seen that if the HALT discrete is failed on, the GPC will be forced to the HALT state even if another discrete is present. This occurs because the IOP performs a system reset and clamps the microcode upon receipt of the HALT discrete. These two events prevent a GPC from executing any software. Therefore, a GPC with a HALT discrete failed on is not usable as PASS or BFS.

A PASS GPC which has a HALT discrete failed off should not be powered off in RUN. If the GPC is powered off in RUN without a HALT discrete, a putaway sequence will be performed which will save the exact state of the GPC at the time of powerdown. If the HALT discrete is later recovered and the GPC is brought back up, the GPC will attempt to perform GPC initialization and then join the CS, but will most likely not be able to join because of the putaway that was done at the GPC powerdown. A re-IPL of the GPC is required to ensure proper recovery.

For GPCs that are powered down in STBY, all the machine putaway that is ever going to be done has been done and the switch monitor sequence is now running. If this GPC's HALT discrete can be recovered, it can be reactivated with no problems whatsoever.

A powered down GPC which loses its HALT discrete cannot be reactivated. When power is reapplied, a power on reset is done which clears the HALT/STDBY/RUN discretes. The GPC then expects the HALT discrete to be reset which will halt the microcode during initialization. Since the HALT discrete is not present, the system continue to attempt to initialize and will access the power on reset PSW, which is coded (in PASS) to put the GPC

in the WAIT state. Recovery, of a PASS GPC, is possible if the HALT discrete can be momentarily reset while GPC power is being applied. The HALT discrete can then be lost and the GPC can be brought up through STDBY since the GPC latched its last state and believed that it was seeing a HALT discrete.

2.4.5.3 STBY Discrete Failure

The STBY discrete failed on has no impact. HALT has priority over STBY since HALT is a hardware controlled state. RUN has priority over STBY in the PASS, BFS, and IPL software. Therefore, a GPC with a failed on STBY discrete may be used in a nominal manner.

A PASS GPC with the STBY discrete failed off cannot be moded to HALT. This is because the PASS requires software cleanup in STBY when transitioning from RUN to HALT. If the failed off STBY discrete is not noticed before the GPC is moded to HALT, the GPC will have to be re-IPL'd. IPL will work by taking the GPC directly to RUN from HALT in the IPL procedure. It is not possible to HISAM dump a GPC that has a failed off STBY discrete. If the STBY discrete fails off in the BFS, it may still be moded to HALT. The only BFS impact is that it can only be operated in RUN and will therefore have control of the payload buses.

2.4.5.4 RUN Discrete Failure

The RUN discrete failed on in a GPC will result in the GPC's being in RUN when the mode switch is taken to STBY. In the PASS this means that the GPC should not be taken to HALT because the cleanup which occurs in STBY cannot be performed. The only impact to the BFS will be that it can only be operated in RUN and will therefore have control of the payload buses.

With the RUN discrete failed on, a GPC can be IPL'd.

The RUN discrete failed off will make a PASS GPC unusable when the mode switch is taken from RUN to STBY. If the RUN discrete fails off in a PASS GPC which is not in RUN, then it can never be moded to RUN.

The RUN discrete failed off in the BFS will prevent it from being taken to RUN once its mode switch has been taken out of RUN. However, the BFS can be operated, and, if necessary, engaged, in STBY. The only impact will be that preengaged and in STBY, the BFS will not control the payload buses.

A GPC with a failed off RUN discrete can be HISAM dumped. It can also be IPL'd, but only as BFS.

2.5 REFERENCES

Data Processing Subsystem Principles of Operation. Rockwell SD 74-SH-0230C, April, 1978

IBM System/4PI Model AP-101 C/M Principles of Operation. IBM No. 6246156B, January 30, 1979

GPCIPL User's Guide. Rockwell MG077202, Dec. 3, 1982

Backup System Services PRD. Rockwell MG038101

CPU/IOP Automated Logic Diagrams (ALD). IBM

GPC Specification. Rockwell MG615-001

Mass Memory Unit Integrated Computer Systems. MPRN-9, June 26, 1983

Software Awareness Memo No. 28

MULTIFUNCTION CRT DISPLAY SYSTEM

4

4 MULTIFUNCTION CRT DISPLAY SYSTEM

SECTION 4
MULTIFUNCTION CRT DISPLAY SYSTEM

CONTENTS

Section		Page
4.1	MCDS OVERVIEW	4-1
4.1.1	Keyboard Unit	4-2
4.1.2	Display Unit	4-2
4.1.3	Display Electronics Unit	4-2
4.1.4	MCDS/DPS Interface	4-3
4.1.5	Crew Controls - PNL O6	4-4
4.1.5.1	Crew Controls - PNL F7, C2	4-5
4.1.5.2	CRT Select, MAJ FUNC Select, and CRT Power Switch	4-6
4.1.6	MCDS Component Locations	4-7
4.1.7	MCDS Physical Characteristics	4-7
4.2	KEYBOARD UNIT	4-8
4.2.1	Features/Performance	4-10
4.2.2	MCDS Keyboard Layout	4-12
4.2.3	Keyboard Unit Connectors	4-13
4.2.3.1	KYBD Connector Pin Layout	4-14
4.2.4	Key	4-15
4.2.4.1	KYBD Switch electronics	4-16
4.3	DISPLAY UNIT	4-17
4.3.1	DEU to DU Analog Signals	4-19
4.3.1.1	X and Y Deflection Signals	4-19
4.3.1.2	Video Signals	4-19
4.3.2	Phosphor Protect	4-19
4.3.3	Power	4-20
4.3.4	BITE	4-20
4.3.4.1	CRT Talkbacks	4-21
4.3.5	DU Features/Performance	4-22
4.3.6	Thermal Considerations	4-23
4.3.7	Display Unit Connectors	4-24
4.3.7.1	DU Connector Pin Layout	4-26
4.4	DISPLAY ELECTRONICS UNIT	4-27
4.4.1	DEU Functional Flow	4-28
4.4.2	DEU Features/Performance	4-30
4.4.3	DEU Physical Layout	4-33
4.4.4	DEU Connectors	4-34
4.4.4.1	DEU Connector Pin Layout	4-38
4.4.5	Power Supply	4-39
4.4.5.1	Power Supply Drawing	4-40
4.4.6	DEU/KYBD Interface	4-43
4.4.7	DEU/DU Interface	4-44
4.4.8	DEU/IOP Interface	4-44
4.4.8.1	MIA	4-44
4.4.8.1.1	Receiving data	4-44
4.4.8.1.2	Transmitting data	4-45
4.4.8.2	MIA Adapter	4-45
4.5	HIP POCKET IFM PROCEDURES	4-46
4.6	MCDS OPERATIONS	4-48
4.6.1	GPC/MCDS Data Communications	4-48

Section Page

4.6.1.1 GPC POLL Detail 4-49
4.6.1.1.1 MSG 1 - MCDS STATUS REQUEST 4-50
4.6.1.1.2 MSG 2 - IPL BITE STATUS REQUEST 4-51
4.6.1.1.3 MSG 3 - RESET SPL 4-51
4.6.1.1.4 MSG 4 - BUFFER FILL 4-52
4.6.1.1.5 MSG 5 - TIME FILL 4-53
4.6.1.1.6 MSG 6 - DISPLAY DATA FILL 4-54
4.6.1.1.7 MSG 7 - FORMAT DATA FILL 4-55
4.6.1.1.8 MSG 8 - DEU MEMORY DUMP 4-56
4.6.1.2 POLL RESPONSE Detail 4-57
4.6.1.2.1 DEU Control Program (DCP) timing windows that affect poll response 4-58
4.6.1.2.2 MCDS status response 4-61
4.6.1.3 DEU Status Words 4-62
4.6.1.3.1 HARDWARE STATUS WORD 1 4-63
4.6.1.3.2 HARDWARE STATUS WORD 2 4-65
4.6.1.3.3 Software status word 4-66
4.6.2 MCDS Displays 4-67
4.6.2.1 SPEC/ROLL-IN DISP Restrictions 4-67
4.6.2.2 SPEC/DISP Allocations (Revised) 4-69
4.6.2.2.1 General notes 4-70
4.6.2.2.2 Sources 4-70
4.6.3 DEU IPL 4-71
4.6.4 SASTP 4-77
4.6.5 OTP 4-81
4.6.5.1 OTP Functions 4-82
4.6.5.2 DEU Memory Dump - Applications 4-84
4.6.6 MCDS Messages 4-85
4.6.6.1 DEU Annunciated Messages 4-86
4.6.6.2 GPC Annunciated Error Messages 4-87
4.6.7 Console Tools for Determining CRT/DEU Assignments 4-90
4.6.7.1 DK Transmitter Enables 4-90
4.6.7.2 BFC CRT Switch Positions 4-91
4.6.7.3 DEU ON/STANDBY 4-92
4.6.8 Keyboard Inputs 4-94
4.6.9 Keyboard Syntax 4-98
4.6.10 DK Listen 4-100
4.6.11 CRT Assignments 4-103
4.6.11.1 CRT MF Switch Logic 4-105
4.6.11.2 GPC/CRT Keyboard Entries 4-108
4.6.11.3 Memory Configuration Change 4-112
4.6.11.4 SPEC/ROLL-IN DIST Restrictions 4-115
4.6.11.5 BFC CRT DISPLAY/SELECT Switch Selections 4-117
4.6.12 Uplink - DEU Equivalent 4-119
4.6.13 Roll-in Display Processing 4-121
4.6.14 MCDS Parameters 4-129
4.6.14.1 D/L Parameters 4-129
4.6.14.2 MCDS - MCC Special Comps 4-133
4.7 MCDS SOFTWARE FLOWS 4-134

DRAWINGS

Drawing		Page
4.7.1	PASS MCDS Input Processor (DMI MCDS IN)	4-135
4.7.2	PASS MCDS Message Processor (DMM MCDS PROCESS)	4-137

SECTION 4
MULTIFUNCTION CRT DISPLAY SYSTEM

4.1 MCDS OVERVIEW

The Multifunction CRT Display System (MCDS) provides for man-machine interface between Space Shuttle crewmembers and the Orbiter's Data Processing System (DPS). Primary crew/system interaction is through a keyboard and a Cathode-Ray Tube (CRT) display. Display background formats are fixed, with all of the various display types determined premission and placed in Display Electronics Unit (DEU) memory, General Purpose Computer (GPC) memory, or on the Mass Memory (MM). Foreground dynamic parameters, as a result of real-time computations, are provided to the DEU by the GPC for CRT display.

The MCDS consists of three major units:

A. Keyboard Unit (KYBD unit) for operator inputs.

B. Display Unit (DU or CRT) consisting of a 5- by 7- inch CRT display.

C. DEU which interfaces with the KYBD unit, DU, and the GPC.

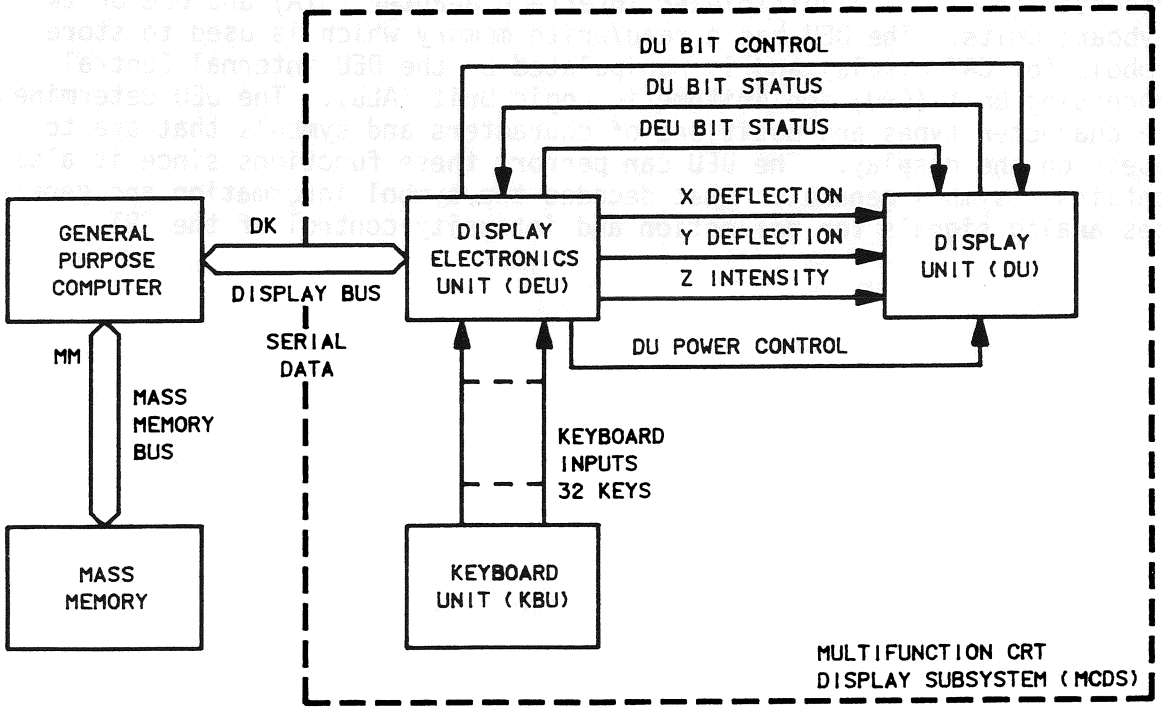


Figure 4-1.- MCDS functional block diagram.

4.1.1 Keyboard Unit

The keyboard unit consists of a set of 32 keys that provide information to the GPC via the DEU. This device allows crewmember to select CRT displays, respond to caution and warning conditions, and command certain actions to be taken by the data processing system.

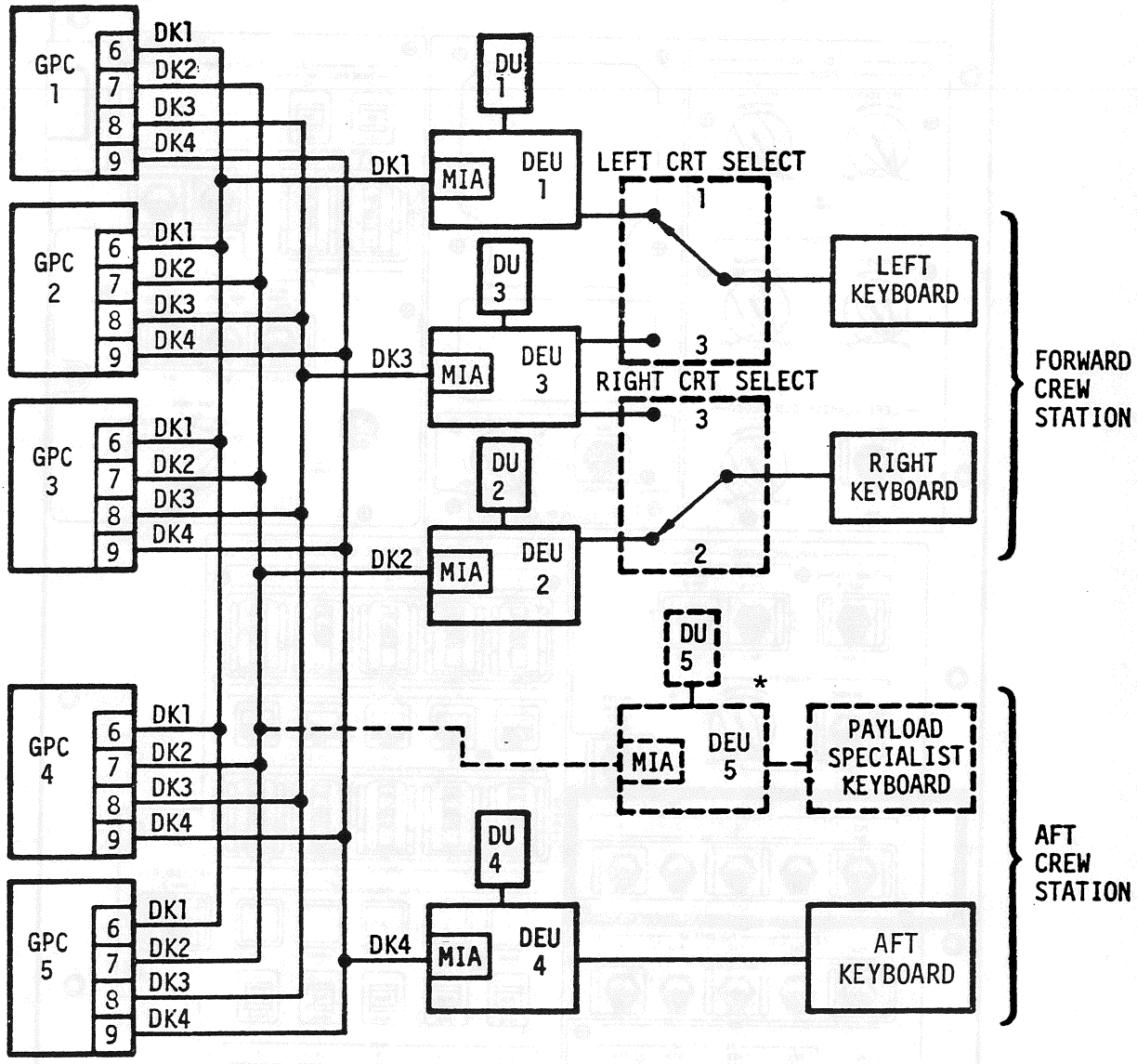
4.1.2 Display Unit

The display unit uses a magnetic-deflected, electrostatic-focused CRT. The unit includes two identical deflection amplifiers, a video amplifier, low-voltage power supply, and a high-voltage power source. When supplied with X and Y deflection signals and video inputs from the DEU, the CRT (DU) will display alphanumeric and graphic (vector, circle) information. Characters can be flashed and the CRT brightness varied for individual characters. The CRT has a single color phosphor - green.

4.1.3 Display Electronics Unit

The DEU interprets the keyboard and GPC information to provide proper deflection signals to the CRT. It interfaces with the GPC's Input/Output Processor (IOP) via a Multiplexer Interface Adapter (MIA) and one or two keyboard units. The DEU has a read/write memory which is used to store symbols for CRT display and is manipulated by the DEU internal Central Processing Unit (CPU) and Arithmetic Logic Unit (ALU). The DEU determines the character types and positions of characters and symbols that are to appear on the display. The DEU can perform these functions since it also contains a symbol generator that decodes the symbol information and generates analog signals for deflection and intensity control of the CRT.

4.1.4 MCDS/DPS Interface (Reference Space Shuttle System Handbook drawings 8.1 and 8.6)



* Not currently implemented

Figure 4-2.- MCDS/DPS interface.

4.1.5 Crew Controls - PNL O6 (Reference SSSH section 20)

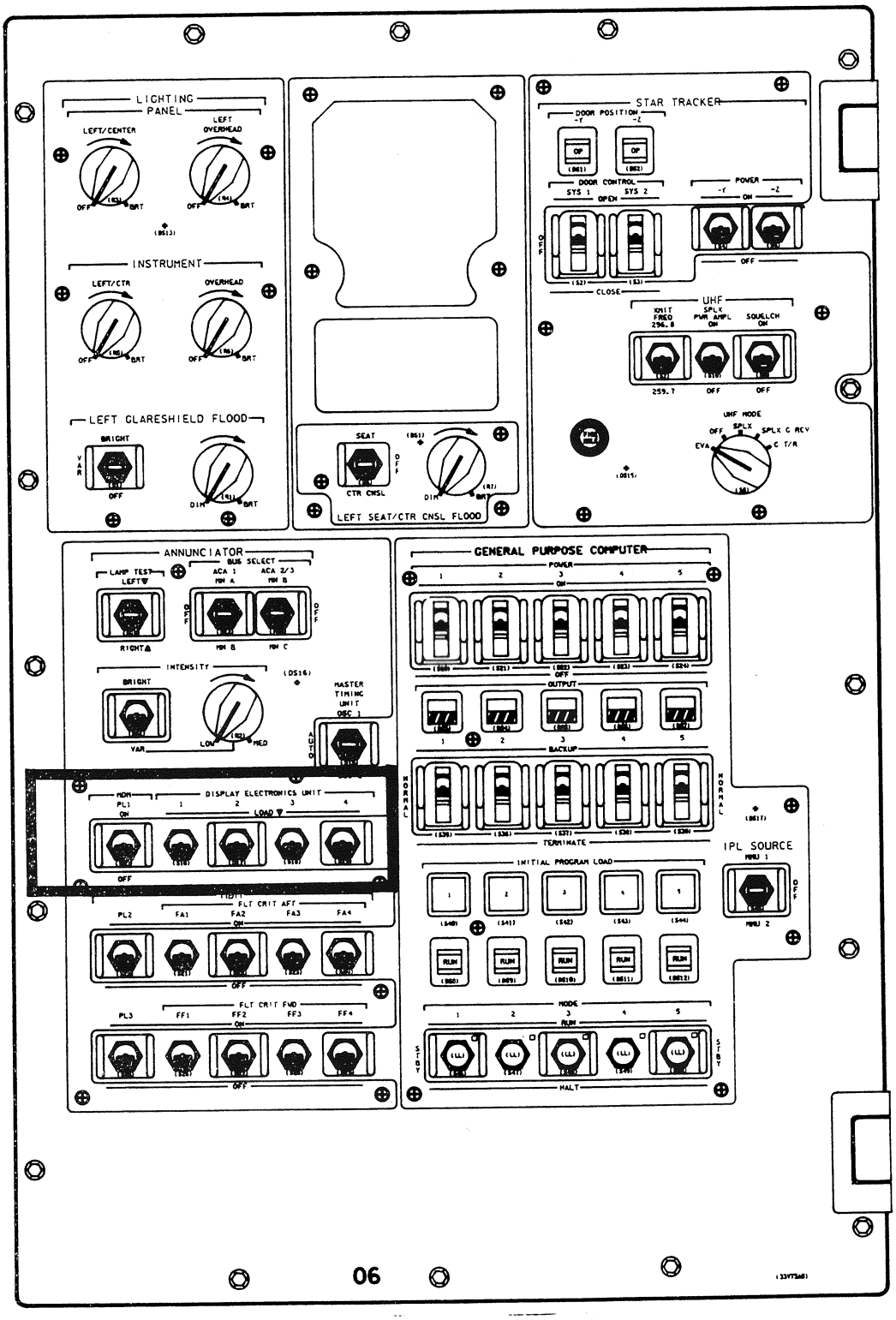


Figure 4-3.- Panel 06.

4.1.5.1 Crew Controls - PNL F7, C2 (Reference SSSH section 20)

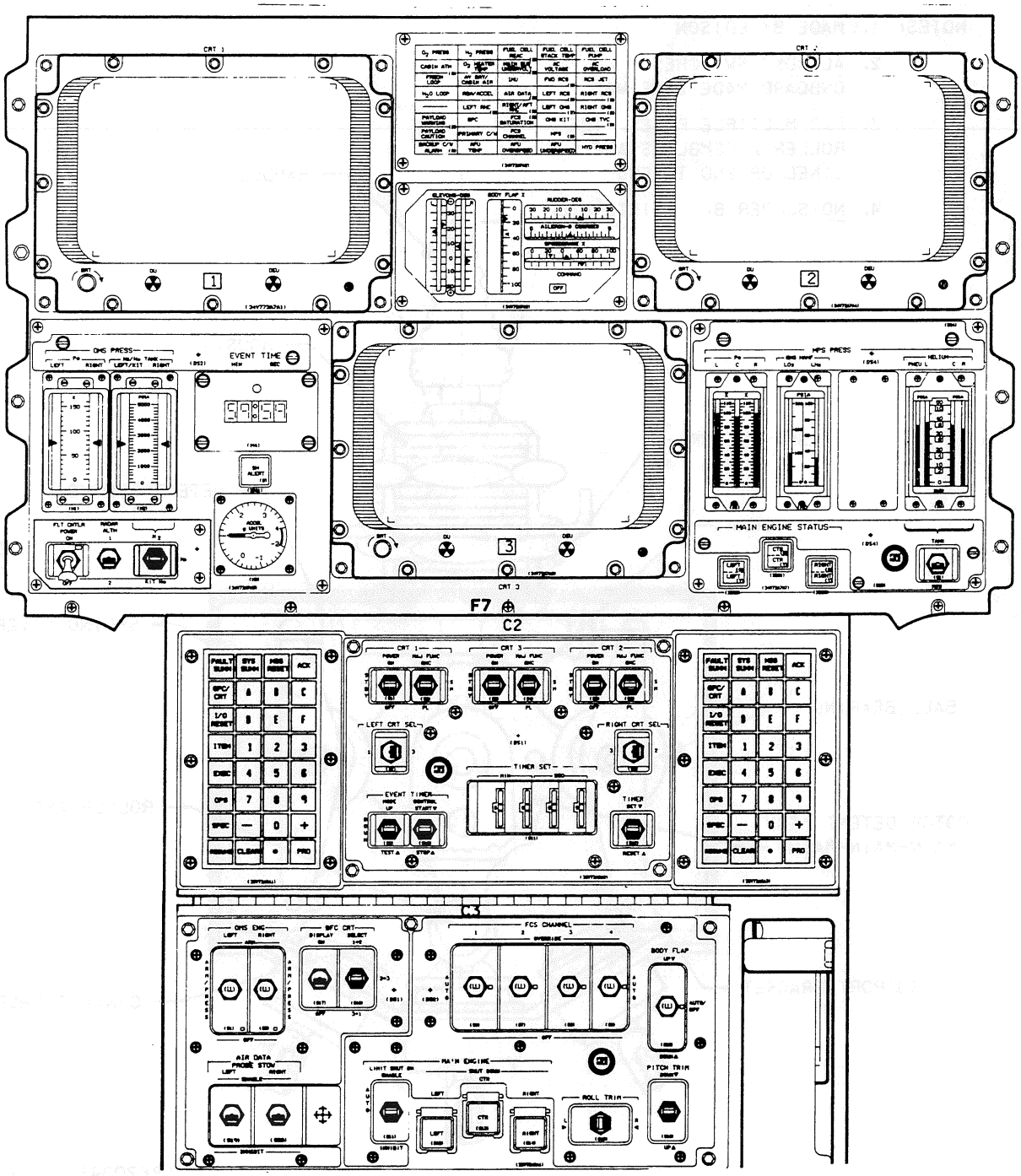


Figure 4-4.- Flight deck, forward cockpit - panels F7 and C2.

4.1.5.2 CRT Select, MAJ FUNC Select, and CRT Power Switch

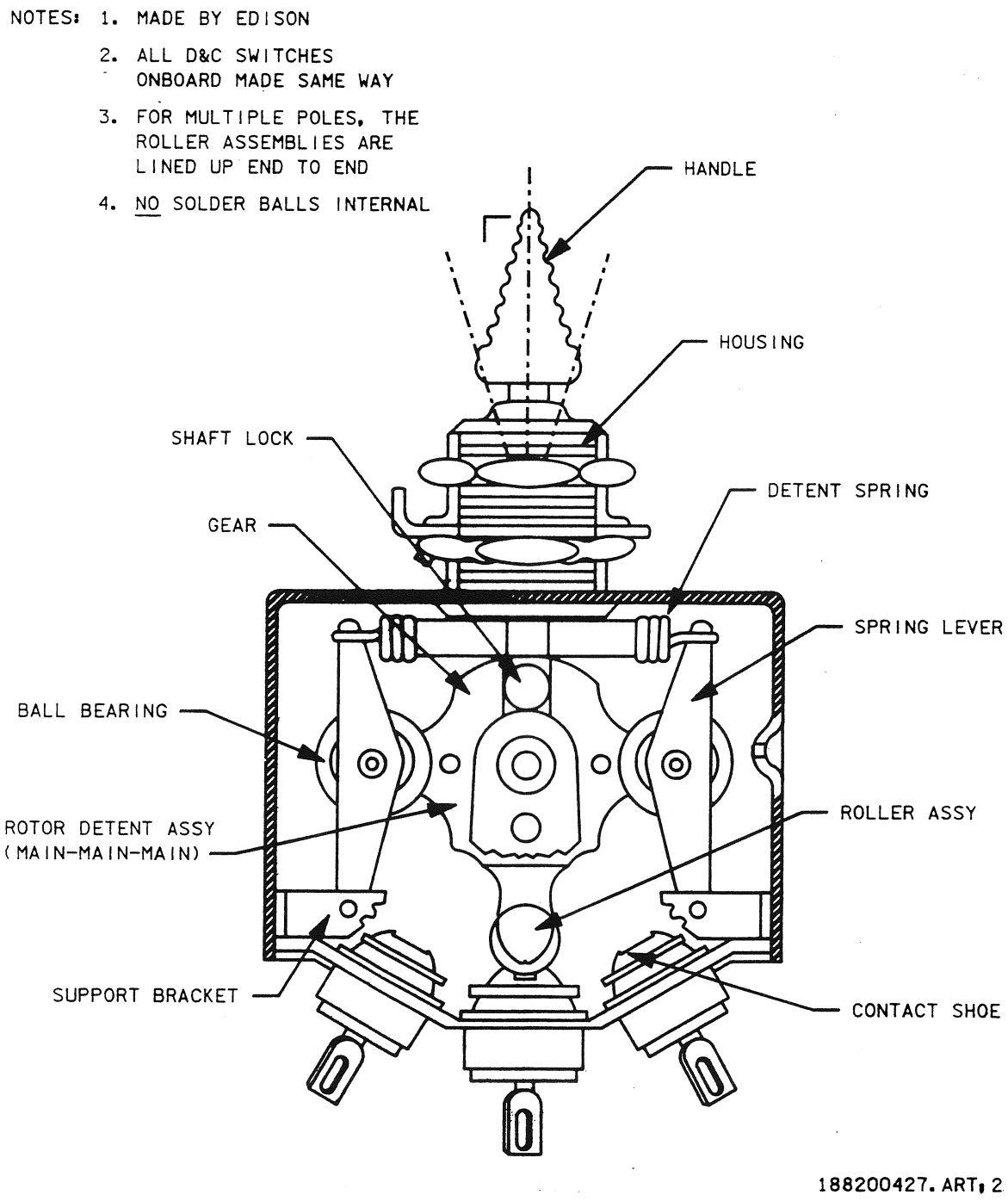


Figure 4-5.- Toggle switch internal assembly drawing.

4.1.6 MCDS Component Locations

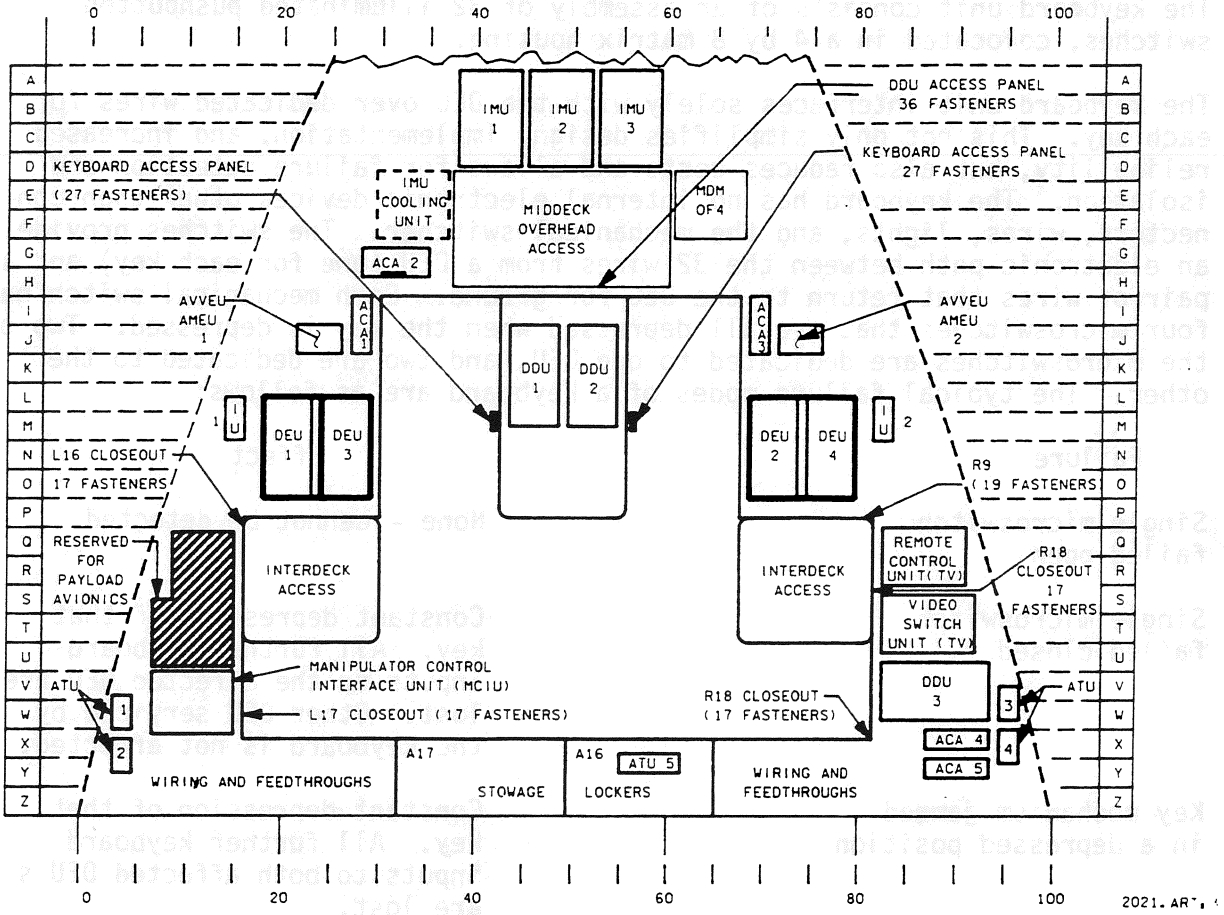


Figure 4-6.- Component locations - flight deck.

4.1.7 MCDS Physical Characteristics

	Size (inches) L x W x H	Weight (lb)	Power (W)
DEU	18.0 x 10.12 x 7.62	35.70	206.0 (26 x 51 "M" format)
Keyboard	6.25 x 4.62 x 7.36	5.79	5.0
DU	16.30 x 10.26 x 7.4	27.00	115 at 75% duty cycle

4.2 KEYBOARD UNIT

The keyboard unit consists of an assembly of 32 illuminated pushbutton switches, colocated in a 4 by 8 matrix housing.

The keyboard unit interfaces solely with the DEU over dedicated wires for each key. This not only simplifies design, implementation, and increased reliability, but also reduces costs and allows for failure detection and isolation. The keyboard has no internal electronic devices other than connectors, wires, lights, and the mechanical switcher. The switches provide an electronic path between the 32 wires from a DEU (one for each key) and a pair of wires that return to the DEU for ground. Each mechanical switch has four microswitches that are all depressed when the key is depressed. Two of the microswitches are dedicated to one DEU, and two are dedicated to the other. The typical failure modes of a keyboard are as follows:

Failure	Effect
Single microswitch failed open	None - cannot be detected
Single microswitch failed closed	Constant depression of that key. All further keyboard inputs to the affected DEU are lost. Other DEU serviced by the keyboard is not affected.
Key mechanism jammed in a depressed position	Constant depression of that key. All further keyboard inputs to both affected DEU's are lost.
Key mechansim jammed in a non-depressed position	Failed key is lost to both DEU's. All other keys are functional.

Keyboard key illumination is provided via 5 volts, ac Orbiter power supply and is not required for the keyboard to be functional.

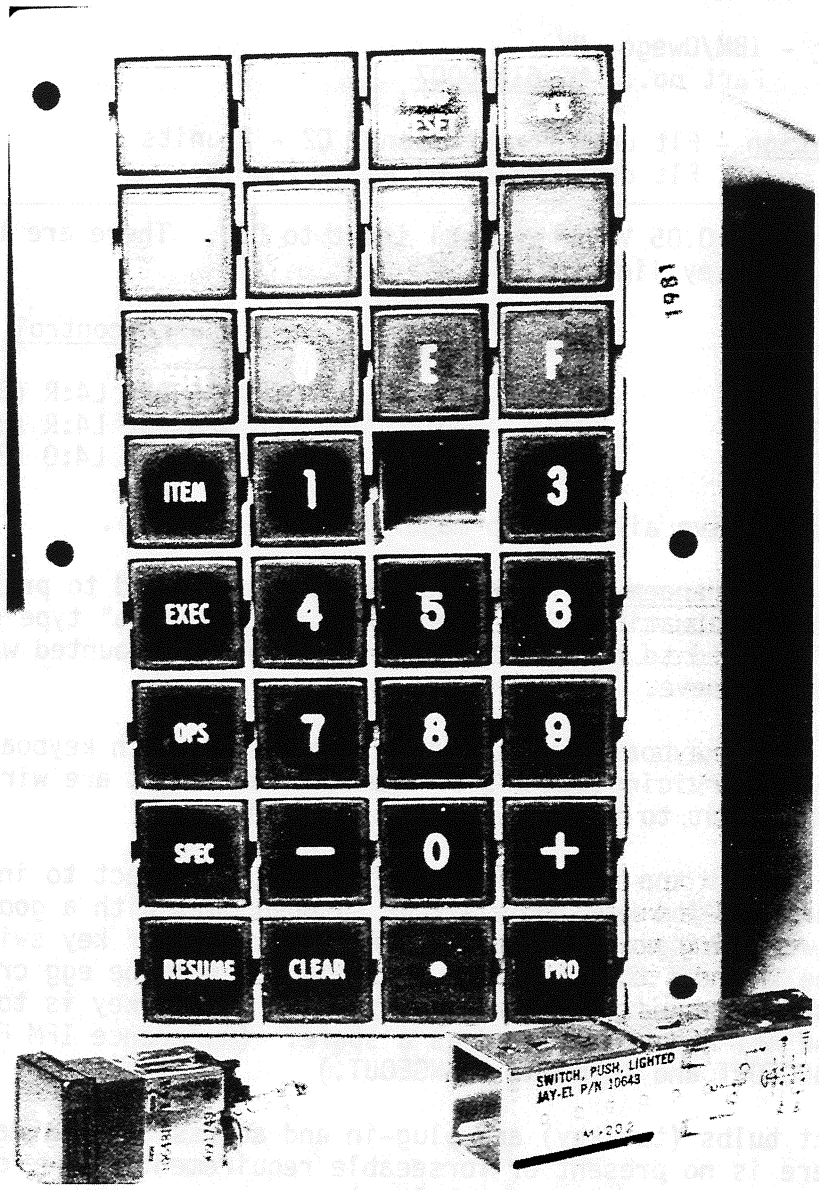


Figure 4-7.- Orbiter keyboard unit.

4.2.1 Features/Performance

Manufacturer - IBM/Owego, NY
Part no.: MC 615-0007

Orbiter location - Flt deck forward/panel C2 - 2 units
Flt deck aft/panel R12 - 1 unit

Power - 5 volts (±0.05 V) for signal input to DEU. There are two ac buses for KYBD key lighting:

Ac bus	KYBD	PNL	Ac source/control
Ac 1 (ΦA)	RH	C2	CB 126 PNL L4:R (3A)
Ac 3 (ΦB)	LH	C2	CB 125 PNL L4:R (3A)
Ac 3 (ΦB)	MS	R12	CB 106 PNL L4:Q (3A)

Cooling - Convective airflow (no major heating concern).

Separation and arrangement - Each pushbutton is mounted to prevent inadvertent actuation of adjacent keys. An "Apollo" type egg crate housing is used to hold each pushbutton which is mounted within a protective sleeve.

Redundancy - The keyboard unit is dual redundant. Each keyboard is capable of servicing two DEU's. Pairs of contacts are wired together for each output to achieve contact redundancy.

Maintenance - Both the keyboard and its keys are subject to in-flight maintenance. The keyboards may be swapped out, with a good aft KYBD unit always being moved forward. Also, in case of key switch failure, both the key and its sleeve may be removed from the egg crate housing for inspection and replacement purposes. If the key is to be changed out, the AFT ACK key is used as a spare. (Reference IFM FDF KEYBOARD KEY CHANGEOUT and KEYBOARD CHANGEOUT.)

NOTE: Light bulbs (two/key) are plug-in and accessible for maintenance. (There is no present or forseeable requirement to perform in-flight maintenance for burnt-out bulbs.)

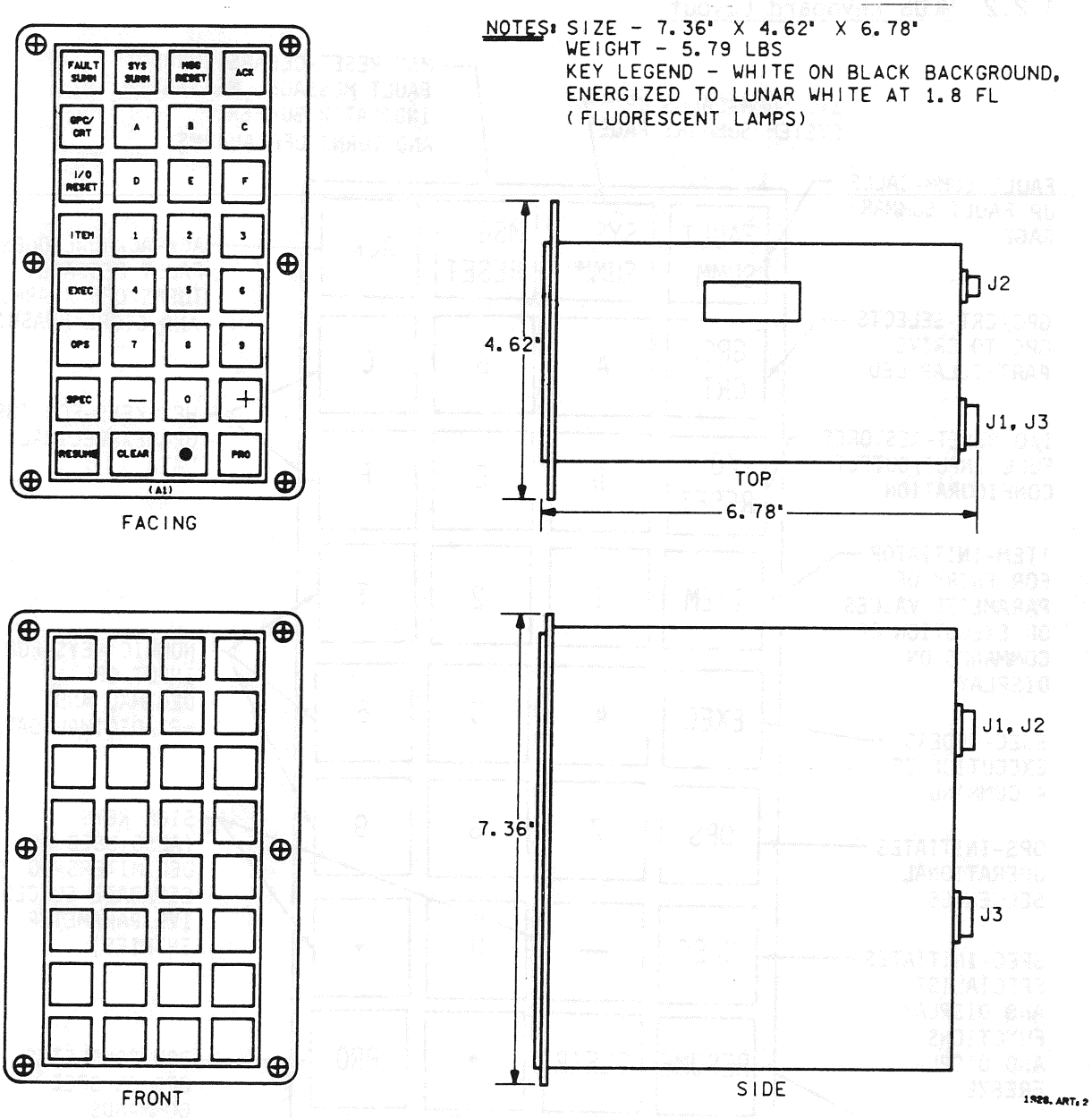


Figure 4-8.- Keyboard unit - physical layout.

4.2.2 MCDS Keyboard Layout

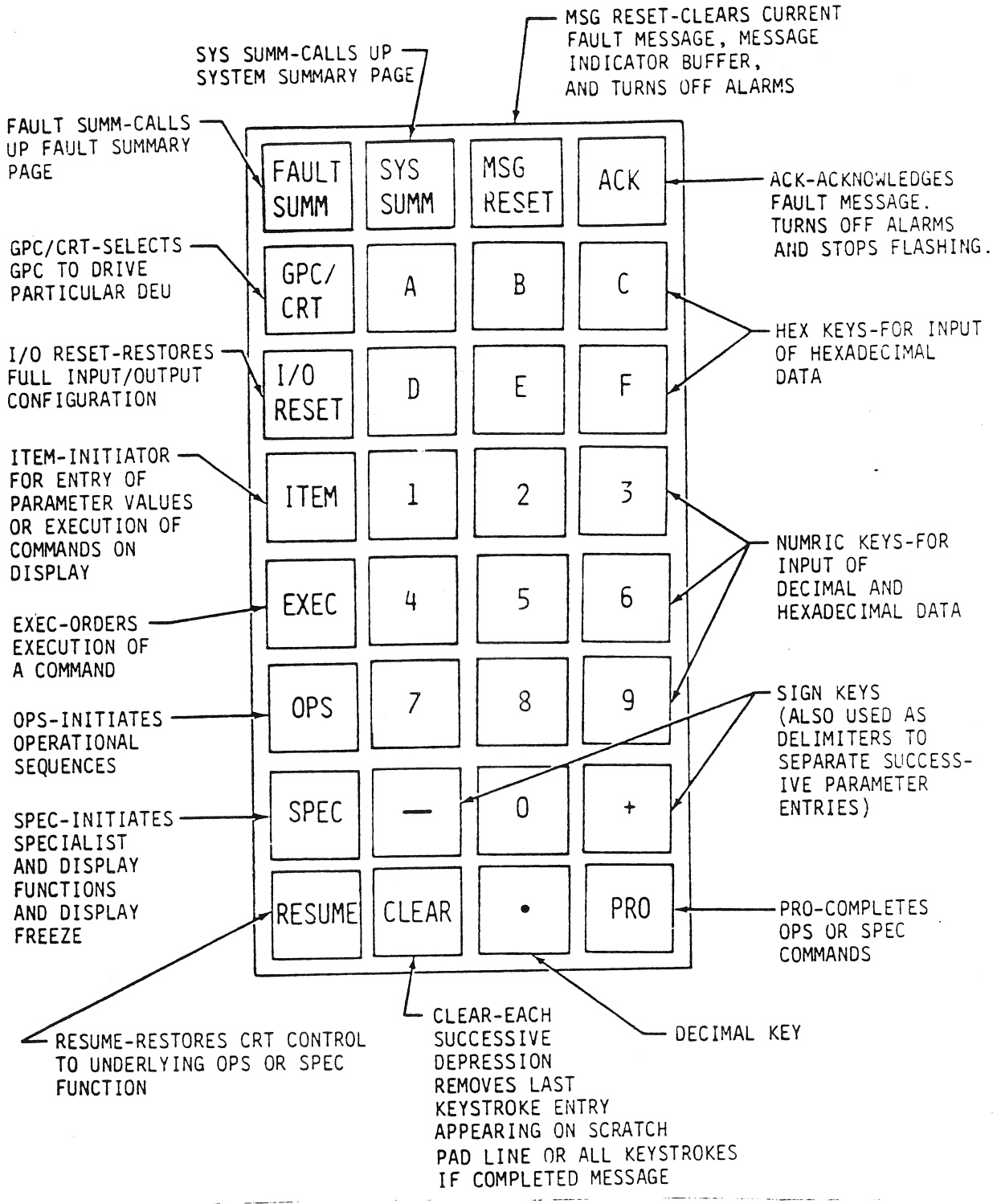


Figure 4-9.- MCDS keyboard.

4.2.3 Keyboard Unit Connectors

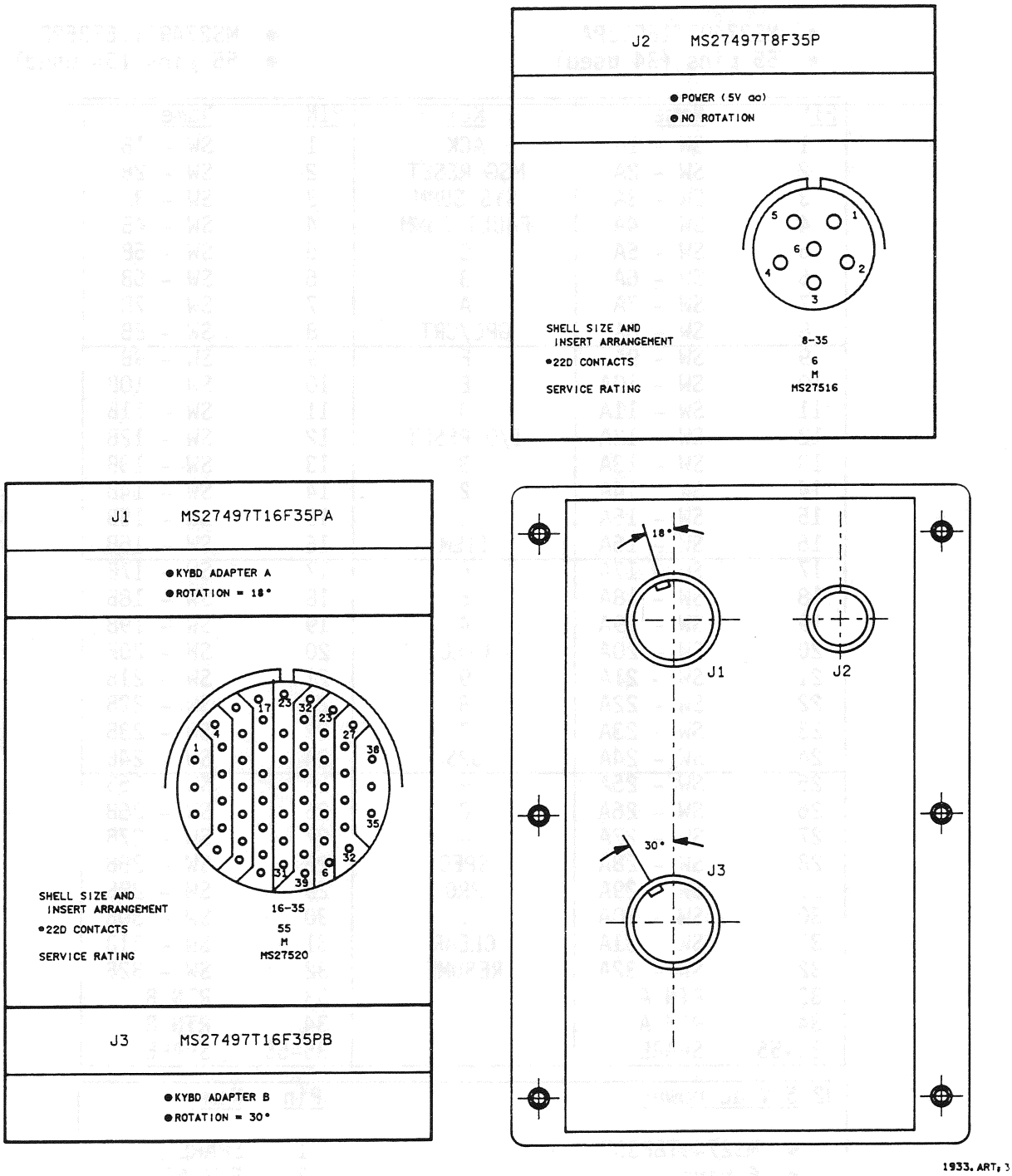


Figure 4-10.- Keyboard unit connectors.

4.2.3.1 KYBD Connector Pin Layout

J1 KB adapter A

- MS27497T16F35PA
- 55 pins (34 used)

J3 KB Adapter B

- MS27497T16F35PB
- 55 pins (34 used)

PIN	Name	KEY	PIN	Name
1	SW - 1A	ACK	1	SW - 1B
2	SW - 2A	MSG RESET	2	SW - 2B
3	SW - 3A	SYS SUMM	3	SW - 3B
4	SW - 4A	FAULT SUMM	4	SW - 4B
5	SW - 5A	C	5	SW - 5B
6	SW - 6A	B	6	SW - 6B
7	SW - 7A	A	7	SW - 7B
8	SW - 8A	GPC/CRT	8	SW - 8B
9	SW - 9A	F	9	SW - 9B
10	SW - 10A	E	10	SW - 10B
11	SW - 11A	D	11	SW - 11B
12	SW - 12A	I/O RESET	12	SW - 12B
13	SW - 13A	3	13	SW - 13B
14	SW - 14A	2	14	SW - 14B
15	SW - 15A	1	15	SW - 15B
16	SW - 16A	ITEM	16	SW - 16B
17	SW - 17A	6	17	SW - 17B
18	SW - 18A	5	18	SW - 18B
19	SW - 19A	4	19	SW - 19B
20	SW - 20A	EXEC	20	SW - 20B
21	SW - 21A	9	21	SW - 21B
22	SW - 22A	8	22	SW - 22B
23	SW - 23A	7	23	SW - 23B
24	SW - 24A	OPS	24	SW - 24B
25	SW - 25A	+	25	SW - 25B
26	SW - 26A	0	26	SW - 26B
27	SW - 27A	-	27	SW - 27B
28	SW - 28A	SPEC	28	SW - 28B
29	SW - 29A	PRO	29	SW - 29B
30	SW - 30A	.	30	SW - 30B
31	SW - 31A	CLEAR	31	SW - 31B
32	SW - 32A	RESUME	32	SW - 32B
33	RTN A		33	RTN B
34	RTN A		34	RTN B
35-55	SPARE		35-55	SPARE

J2 5 V ac power

- MS2749T8F35P
- 6 pins

Pin	Name
1	SPARE
2	5 V AC
3	AC RTN
4	SHLD-RTN
5	CHASSIS RTN
6	SPARE

4.2.4 Key

- Manufacturer

  Jayel Products Inc.
  Gardenia, California

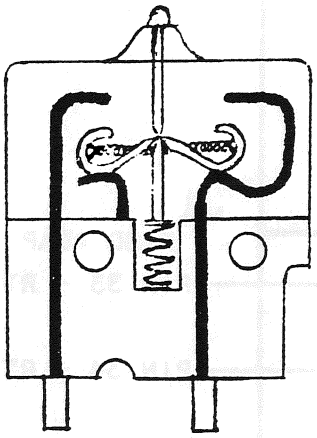
- Part no.

  Jayel - 10772
  IBM - 6091752-2

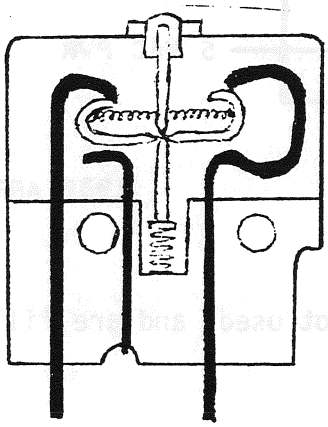
The switch is a watertight 4-pole momentary action device designed for 50,000 cycles.

The key consists of a pushbutton which, when depressed, actuates redundant microswitches.

Microswitch

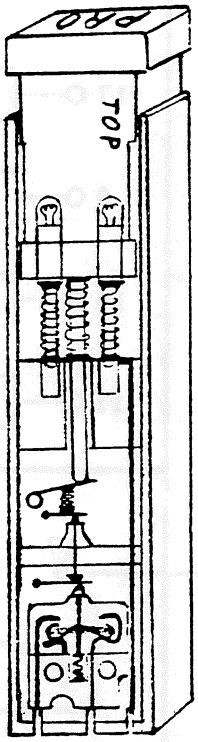


Undepressed

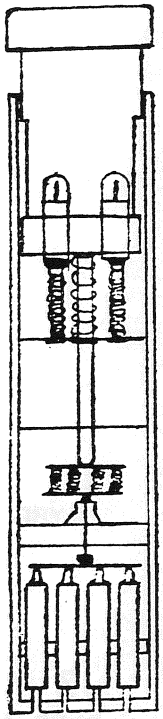


Depressed

Actual size



Top view

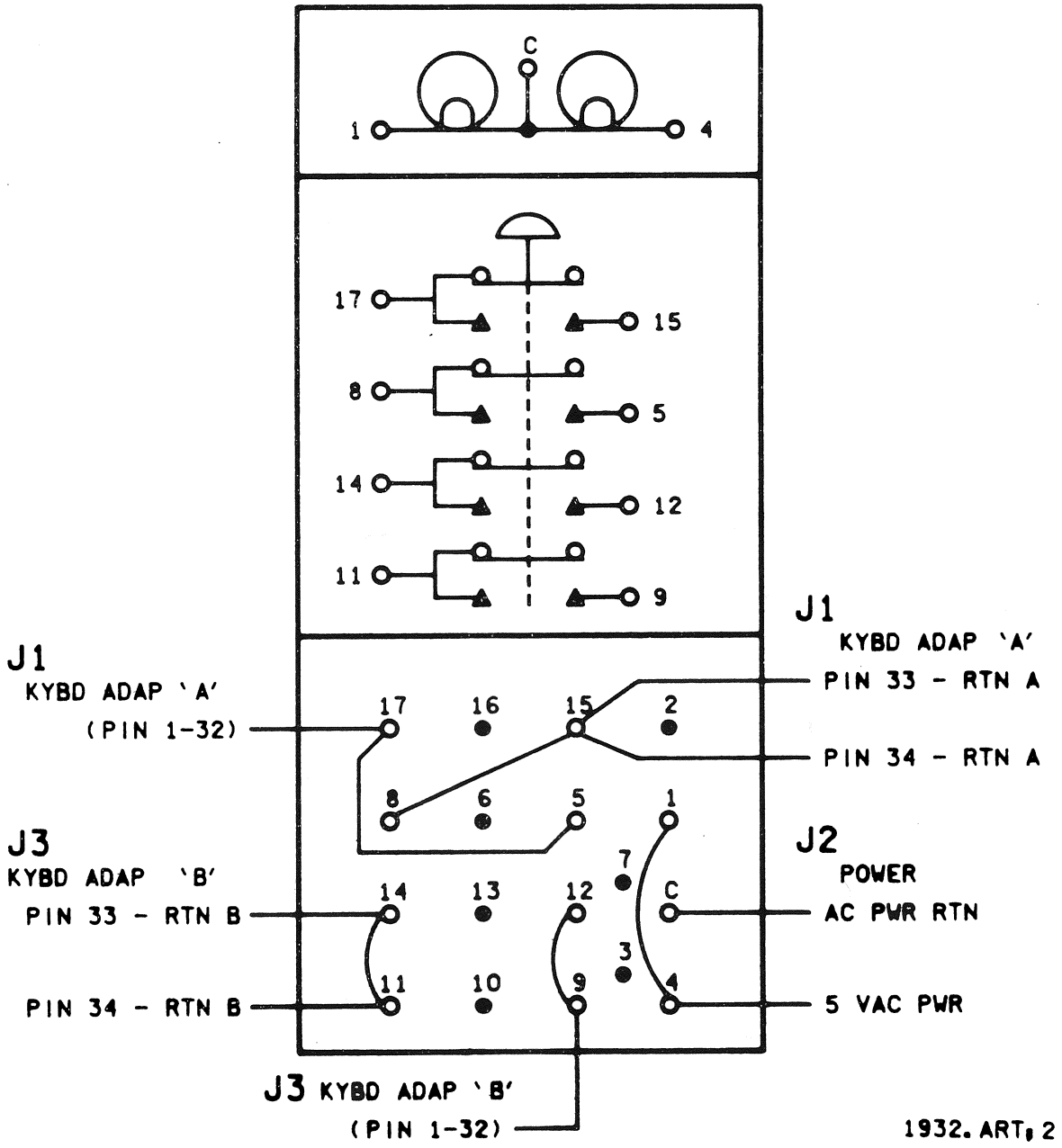


Side view

Key size - 0.74 inch square with key spacing of 0.812 inch C/L to C/L.

NOTE: There are four microswitches per key-switch assembly (see side view) with gold plated contacts. Two of the microswitches are wired in parallel to the J1 connector and two are wired in parallel to the J3 connector. Thus, for any one of the 32 lines from a DEU, there are two micro-switches wired in parallel which provide a path to ground when the key is depressed.

4.2.4.1 KYBD Switch Electronics



NOTE: Switch pins 2, 3, 6, 7, 10, 13, and 16 are not used, and are filled with a nonconductive polyresin.

Figure 4-11.- Switch electronics.

4.3 DISPLAY UNIT

The Display Unit (DU) converts deflection and video signals developed by the DEU into a visual display. The DU front panel provides:

- View surface - 5 by 7 inches
- Brightness control
- Ambient light detector
- BITE status indicators - DU and DEU

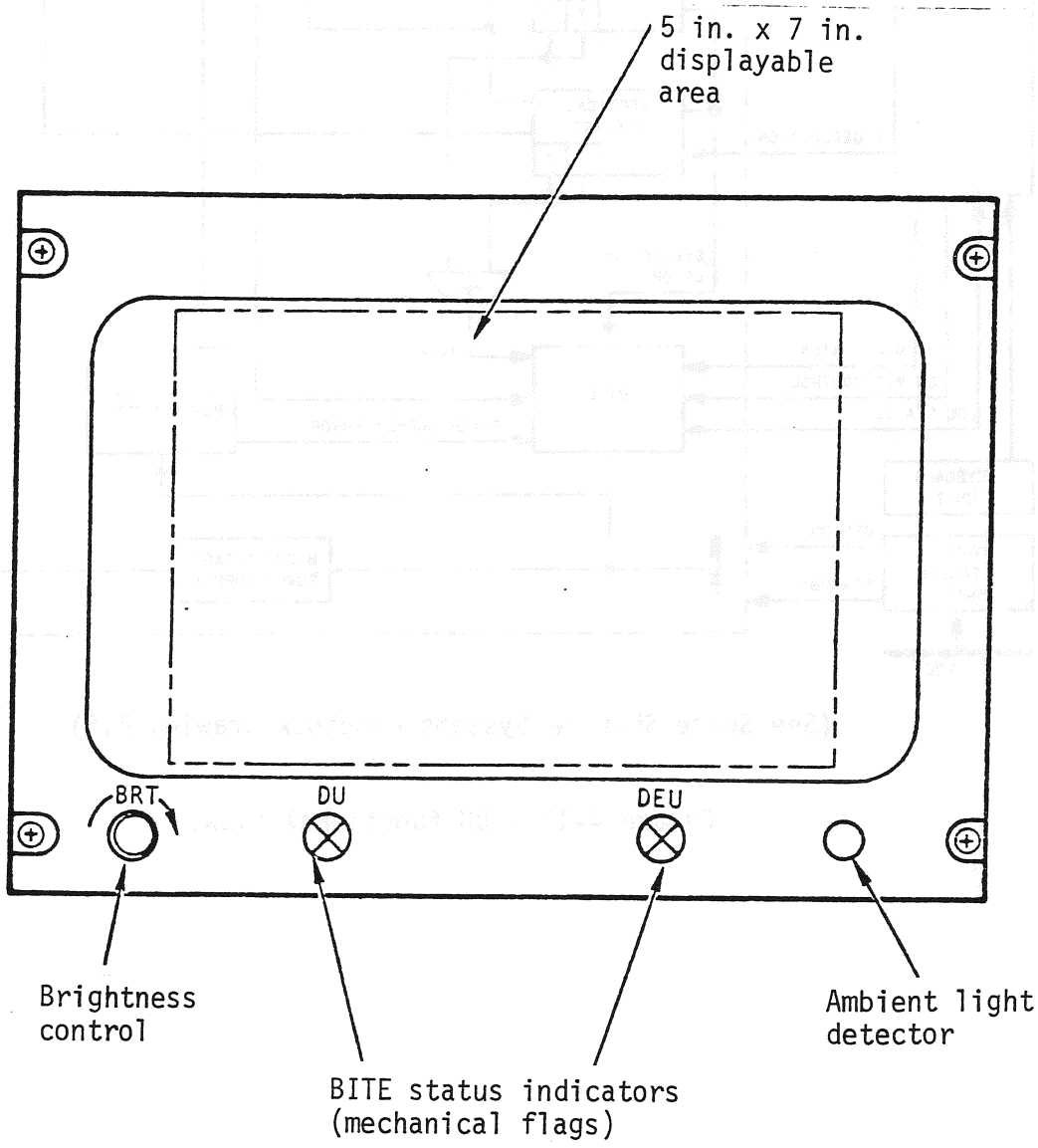
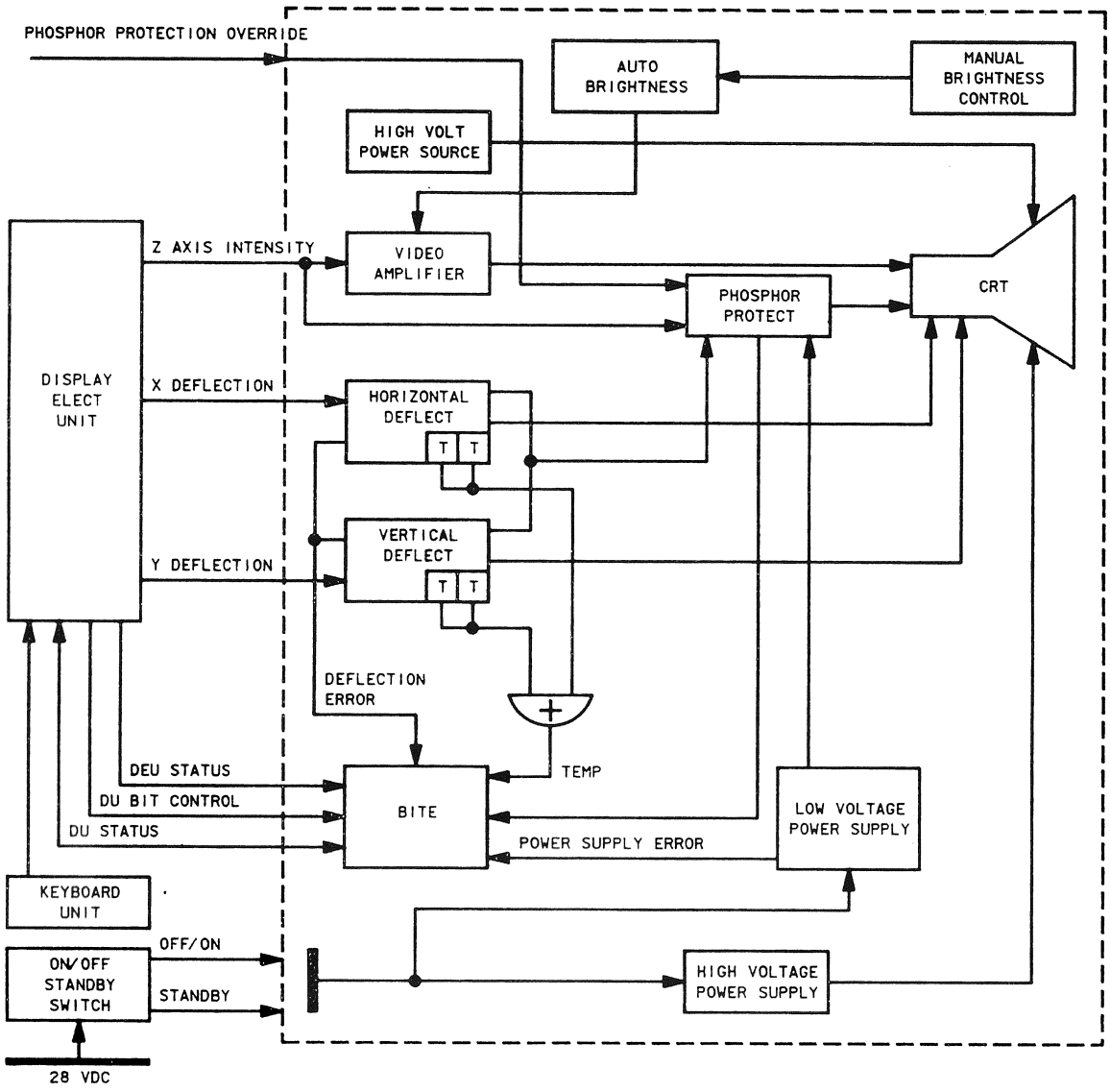


Figure 4-12.- Display unit.



(See Space Shuttle Systems Handbook drawing 8.6)

Figure 4-13.- DU functional flow.

4.3.1 DEU to DU Analog Signals

4.3.1.1 X and Y Deflection Signals

Deflection signals, driven from the DEU symbol generator, are received by the DU and corrected for distortion. The signals are received differentially, scaled, and sent to linear-correction circuitry which rescales them (nonlinearly) so as to correct the signals for geometric distortion inherent in magnetically driven CRT's. From the deflection receivers, the signals are passed to deflection yoke amplifiers. The amplifiers are closed loop voltage-to-current devices whose inputs are analog voltages and outputs result in yoke drive currents. The amplifiers respond to: signals of small amplitude required to create character/symbols and signals of large amplitude required to create certain patterns covering large portions of the CRT screen. The large amplitude signals are also used to position the beam on the CRT screen.

4.3.1.2 Video signals

Video signals, driven from the DEU symbol generator, are received by the DU and consist of an analog blanking signal called Z intensity. From the video receiver, the signal is passed to the video amplifier where it is used to control the CRT beam intensity. The video signal is a linear function of writing speed.

A manual brightness control provides a continuously variable means of setting/adjusting intensity. Also, ambient light conditions in excess of 100-foot lamberts, as sensed by a detector on the face of the DU, cause an automatic brightness control to be input to the video amplifier such that the necessary contrast can be maintained.

4.3.2 Phosphor Protect

To protect the CRT from phosphor "burns" during abnormal conditions, a phosphor protect circuit is incorporated in the DU. The following abnormal conditions will activate phosphor protect circuitry:

- Video signal detected with no X or Y deflections present (small circles can cause this condition, e.g., SPEC 50).
- Execessive voltage detected in video signal
- Loss of, or a steady X or Y deflection current
- Deflection signal detected exceeding 5 percent of 5 by 7-inch viewing range (off-screen beam positioning)
- Power supply error detected

Sensing circuits are used to automatically detect malfunctions and, upon detection, disable the CRT beam current, either by blanking and/or grounding

the control grid - G1. This prevents the possibility of a high-intensity beam current condition that could damage the phosphor. An override of this protect circuitry may be accomplished by means of a jumper on the DU/DEU interface connector (connector J1, pin no. d). This override circuitry is wired in Orbiter DU flight units only. If phosphor protect circuitry is activated, it will result in a DU NO-GO condition indicated by the DU mechanical Built-In Test Equipment (BITE) flag on the front panel of the DU. However, a DU phosphor protect error on vehicle units will not cause the CRT to blank. Also, for vehicle units the phospher protect BITE bit should never be seen because the wire between the DU and DEU associated with phospher protect has been cut and run to ground at the DEU end.

4.3.3 Power

DU components receive power via one of two power supplies, either the low-voltage or high-voltage power supply devices, both of which are internal to the DU itself. Regulated power is distributed by the low-voltage power supply, which receives 28 V dc power from the Orbiter main buses controlled through associated Orbiter control buses. The high-voltage power supply provides necessary voltages to CRT.

DEU	DU	Power source	Controlled via
1	1	MNA FPC 1	CNTL AB1
2	2	MNB FPC 2	CNTL BC2
3	3	MNC FPC 3	CNTL CA1
4	4	MNC FPC 3	CNTL CA2

Power consumption

A DU consumes: 90 W ON
20 W STBY

4.3.4 BITE

BITE monitors performance of DU circuitry continuously to verify operability and to detect/flag DU component failures. BITE signals are generated periodically during the inactive portion of the display cycle as initiated by a BITE timing signal from the DEU. DU status-out of BITE is provided to the DEU as discrete signals via the DU/DEU interface connector - J1. BITE provides failure detection for approximately 96 percent of DU components.

BITE monitors the following DU elements or circuits:

- Video amplifier output
- Temperatures - out of X, Y deflection amplifiers, low-voltage power supply (T >125° C results in error)
- X deflection input and output
- Y deflection input and output
- Phosphor protect output
- Filament current
- Low-voltage power supply

The DU BITE status output to the DEU (via connector J1) includes:

- DU video
- DU deflection
- DU temperature (T >125° C)
- DU phosphor protect - not wired in current Orbiter DEU flight units
- DU filament current
- DU power supplies

4.3.4.1 CRT talkbacks

The CRT talkbacks are electromechanical indicators or flags labeled DU and DEU and are located below the CRT's on each DU. See Space Shuttle Systems Handbook drawing 8.6, MCDS, to determine which BITE errors will cause the flags to trip along with the time delays before trip. The flags are tripped from all black to black and white when an appropriate BITE error is set. There is only one error which will cause the DEU BITE flag to trip that is not reflected in any of the BITE hardware or software registers. This is the DEU power supply error which is set whenever an over or undervoltage is detected and the DEU power supply does not shut down within a required period of time. Note that performance of the Stand-Alone Self-Test Program (SASTP) following CRT powerup is required for any DU BITE error other than temperature error to trip the DU BITE flag. The temperature error will always immediately trip the DU BITE flag. Once tripped, the BITE flags can only be reset manually. To reset these flags, they must be rotated clockwise to the stop then returned counterclockwise to the stop.

4.3.5 DU Features/Performance

Manufacturer - Norden, Part no. MC 615-0006

Orbiter location - Flt deck forward/panel F7 - 3 units
Flt deck aft/panel R12 - 1 unit

Power - 28 V dc

DU	Power source	Controlled via
1	MNA FPC 1	CNTL AB1
2	MNB FPC 2	CNTL BC2
3	MNC FPC 3	CNTL CA1
4	MNC FPC 3	CNTL CA2

Power consumption: 90 W ON
20 W STANDBY

Cooling - Forced air, suction via cabin fans

Size - 7.4 by 10.26 by 16.3 inches

Weight - 27.0 lb

Reliability - 3000 hours MTBF

Maintenance - Display units are remove and replace items. When required during Orbiter flight, a good aft CRT may be swapped out with a bad forward CRT 1 or 2. CRT 3 is not considered for swapout because of interference with the overhead panel ejection T-handle.

CRT - 5 by 7 inches
P43 phosphor
Magnetic deflection
0.012 line width
900 ft phosphor brightness
Electrostatic fixed focus

Writing speed - 13,000 to 80,000 inches per second

Positioning time - X and Y deflection channels, is less than 12 μs for deflection to 99.9 percent of final value.

Vibration - 0.09 G^2/Hz random

4.3.6 Thermal Considerations

Semiconductor junction temperature: Maximum = 125° C, low-voltage power supply (93 transistor maximum) = 109° C

Operating temperature = 20° to 120° F operating range
35° to 120° F ambient

Power dissipation: 102.7 W

Cooling: Type - forced air
Air temp - 35° to 100° F
Flow rate - 0.81 lb/min (48.6 lb/hr) at all inlet temperatures to give a 30° F temp rise in the cooling air
Pressure drop - maximum allowable = 1.50 inches H_2O (0.86 inches H_2O for 0.81 lb/min)

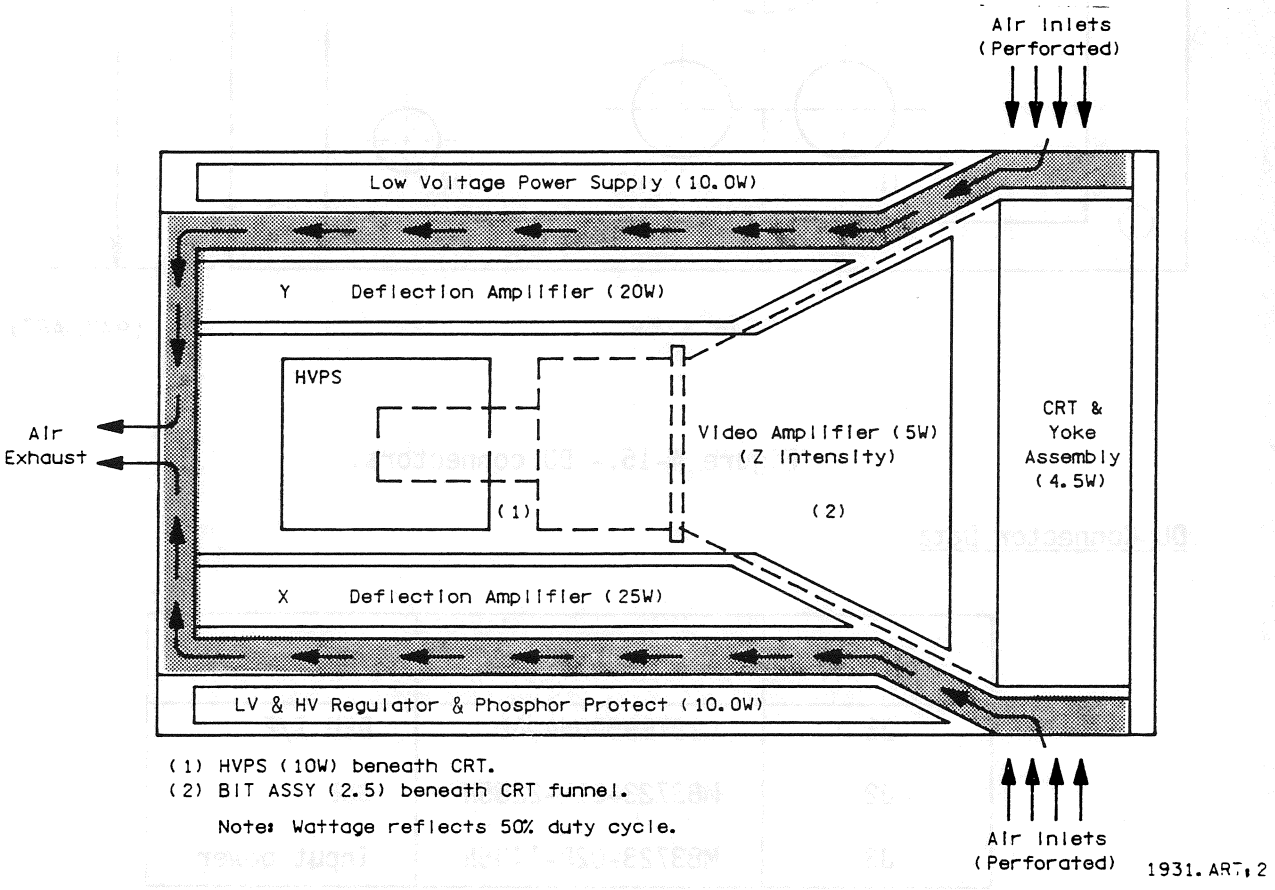


Figure 4-13.- DU air flow.

4.3.7 Display Unit Connectors

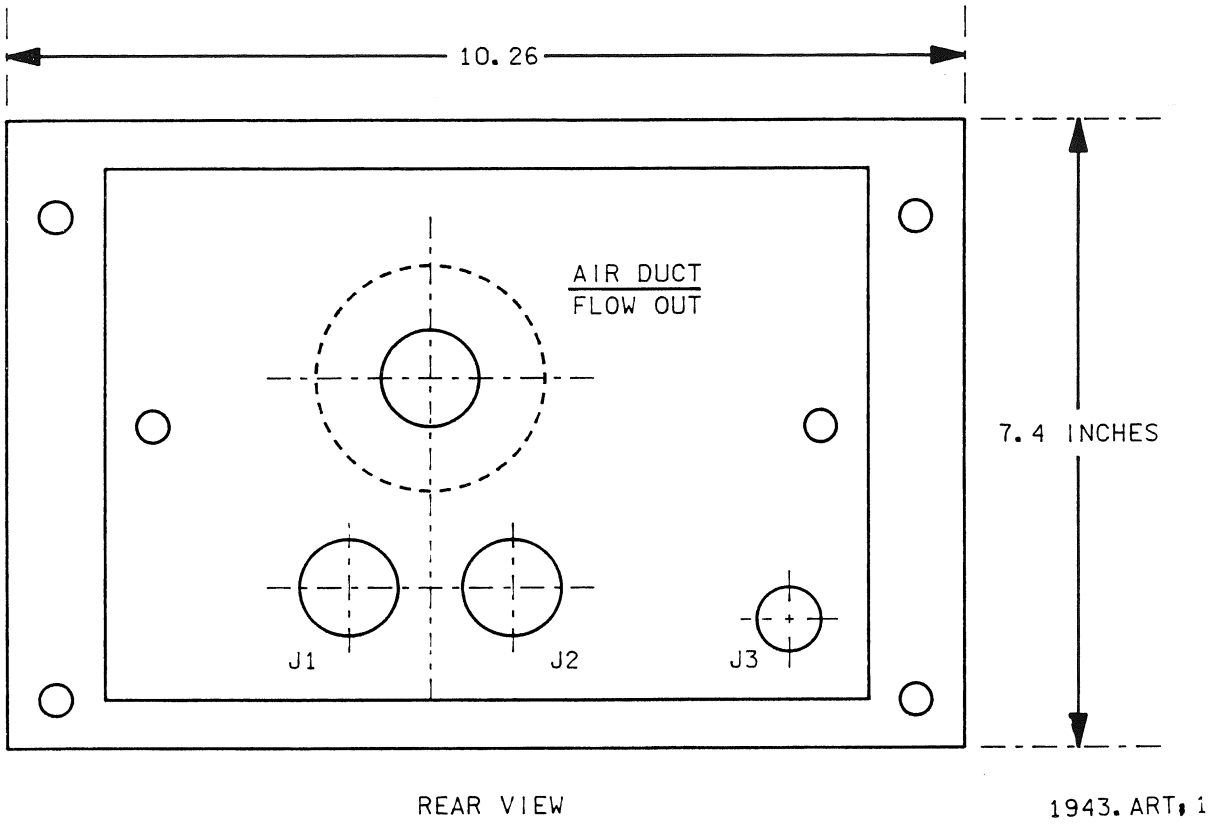


Figure 4-15.- DU connectors.

DU Connector Data

J no.	Part no.	Function
J1	L22TG95PONA-AL	DEU I/F
J2	M83723-01R-2255N	GSE
J3	M83723-02R-1405N	Input power

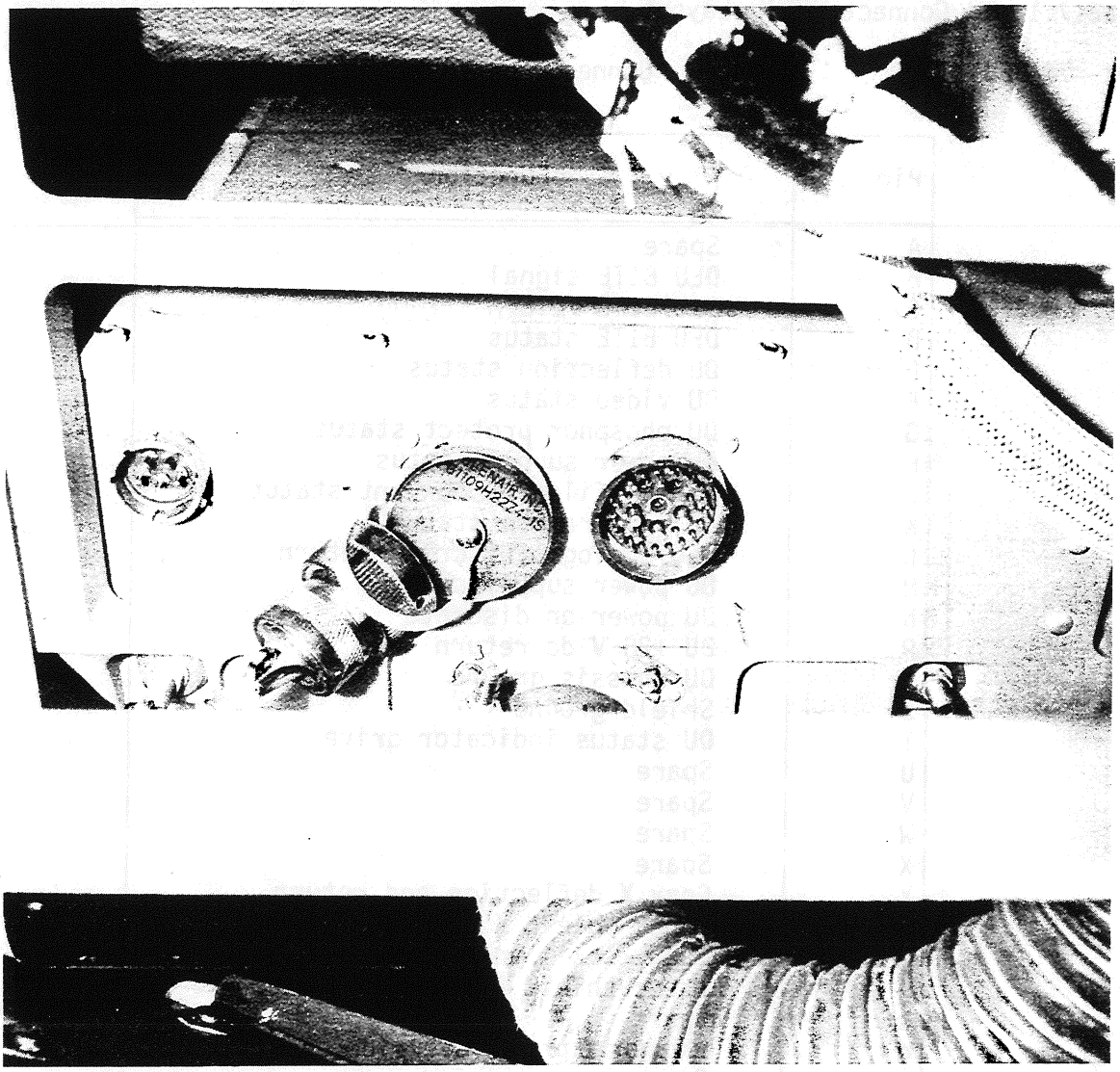


Figure 4-16.- OV-102 Columbia CRT 1.

4.3.7.1 DU Connector Pin Layout

J1 - DU/DEU I/F Connector no.: L22TG95PONA-AL

Pin	Function
A	Spare
B	DEU BITE signal
C	DEU BITE return
D	DEU BITE status
E	DU deflection status
F	DU video status
G	DU phosphor protect status
H	DU power supply status
J	DU CRT filament current status
K	DU temperature status
L	DEU/DU logic discrete return
M	DU power supply common
N	DU power-on discrete
P	DU +28 V dc return
R	DU chassis ground
S	Shield ground
T	DU status indicator drive
U	Spare
V	Spare
W	Spare
X	Spare
Y	Coax X deflection and return
Z	Outer shield X deflection
a	Coax Z-axis and return
b	Outer shield Z-axis
c	Spare
d	Phosphor protect override
e	Spare
f	Phosphor protect override return
g	Coax Y deflection and return
h	Outer shield Y deflection
i	Spare

J2 - GSE (DU checkout) connector no.: M83723-01R-2255N
This connector is not used onboard the Orbiter and is capped.

J3 - Input power connector no.: M83723-02R-1405N

Pin	Function
A	28 V dc power
B	Spare
C	28 V dc power return
D	Spare
E	Chassis ground

4.4 DISPLAY ELECTRONICS UNIT

The Display Electronics Unit (DEU) represents the machine-to-machine management of information for display purposes. It manages keyboard inputs, drives an attached display unit, and responds to inputs/requests from its commanding GPC.

The DEU interprets the keyboard and GPC information in order to provide the proper analog deflection signals to the CRT. It interfaces with the GPC's IOP via an MIA and one or two keyboards via keyboard adapters. The DEU has a read/write memory which it uses to store and manipulate the symbols that are to appear on a given display. It also contains a symbol generator that decodes the symbol information, generating analog signals for deflection and intensity control for transmission to the CRT. The functions of the DEU can be broken down into three basic categories:

A. Interface - IOP MIA, KYBD adapters, power supply, and control switches

B. Information control - via the DEU control program resident in the small processor

C. Display information - via the symbol generator

4.4.1 DEU Functional Flow

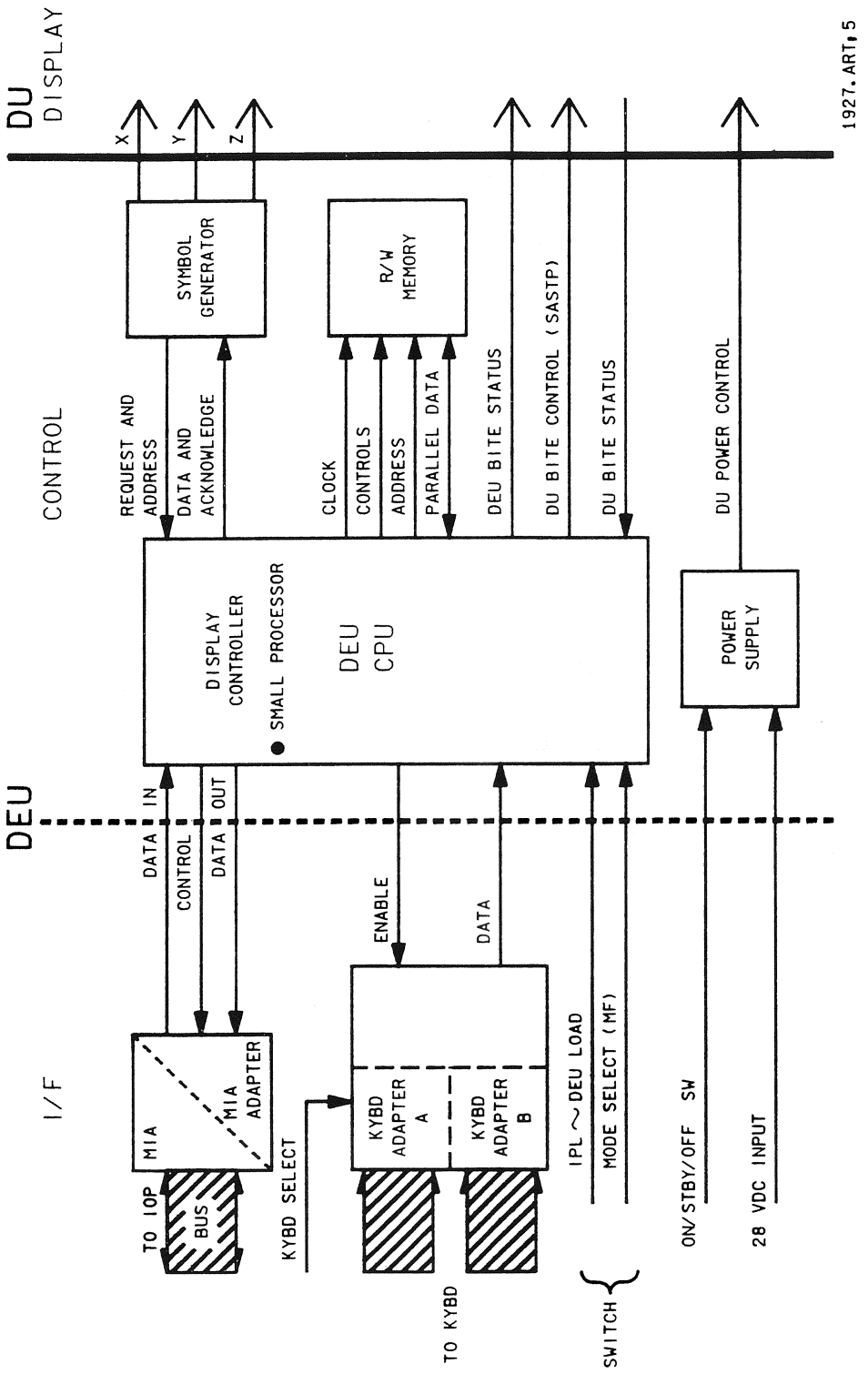


Figure 4-16.- DEU functional flow.

The DEU provides internal control-oriented processing of the following functions:

A. Initialization sequence: DEU LOAD via a Programable Read Only Memory (PROM)

B. MCDS/GPC communication: IOP-to-DEU communications

1. Receives and processes input commands and data

2. Constructs and transmits messages to the GPC (upon demand)

C. SASTP: Stand-Alone Self-Test-Program (GPC menu IPL only)

D. Allocation of DEU Read/Write (R/W) memory (by GPC request)

E. Major function switch (GNC, SM, PL) monitoring. In the event of a failed MAJ FUNC switch, the DEU will transmit the last MF; if there is no last valid MF (i.e., switch failed during DEU IPL), the DEU will transmit the MF GNC.

F. BITE information on DEU as well as the DU

G. Time display: processing of GMT/MET and CRT timer data

H. Error annunciation

1. Automatic response

2. Dedicated fault message line

I. Keyboard inputs

1. Generic syntax checking

2. Display keystrokes on Scratch Pad Line (SPL)

3. Clear miskeyed crew inputs by instruction

4. "Form fill" SPL environment

4.4.2 DEU Features/Performance

Manufacturer - IBM, Owego, NY
Part no.: MC 615-0008

Orbiter Location - Flt deck below PLT and CDR seats (ref. figure 1-5)

Power - 28 V dc

DEU	Power source	Controlled via
1	MNA FPC 1	CNTL AB1
2	MNB FPC 2	CNTL BC2
3	MNC FPC 3	CNTL CA1
4	MNC FPC 3	CNTL CA2

Power comsumption: 202 W ON
0 W STBY

Cooling - Forced air suction via cabin fans.

Size - 7.62 by 10.12 by 18 inches.

Weight - 37.8 lb

Maintenance - DEU's 1 and 3 may be swapped with DEU 4. If DEU 2 fails, the cables for DEU 2 are swapped to DEU 4 (this avoids a physical movement of the boxes for DEU 2 failure) (Reference IFM).

Symbol Generator Data

① Refresh rate - The refresh rate is maintained at 55 frames per second.

② Character generation time - The time for generation of a single character is 8 μs average.

③ Positioning time - The positioning time is program selectable and may be chosen to be 2, 7, or 12 μs.

④ Vector speed - Vectors may be drawn on the CRT screen at the rate of 12.2 μs per inch.

⑤ Circle speeds - There are two circle speeds. For circles from 1-3/4 to 7 inches in diameter, the speed of construction is 20,000 to 80,000 inches per second. For circles 1-3/4 to 0.3 inch in diameter, the speed is 80,000 to 13,000 inches per second, respectively.

⑥ Circle sizes - Circle sizes may be constructed at any point on the display field. The circle radii are variable from 7 to 0.3 inch in diameter.

Although smaller circles may be drawn, the DU writing speed for 0.3 inch in diameter is approximately 13,000 inches per second, the minimum speed at which the circle will remain in focus.

⑦ Character sizes - Characters may be displayed in either of two sizes: 0.125 inch high or 0.150 inch high.

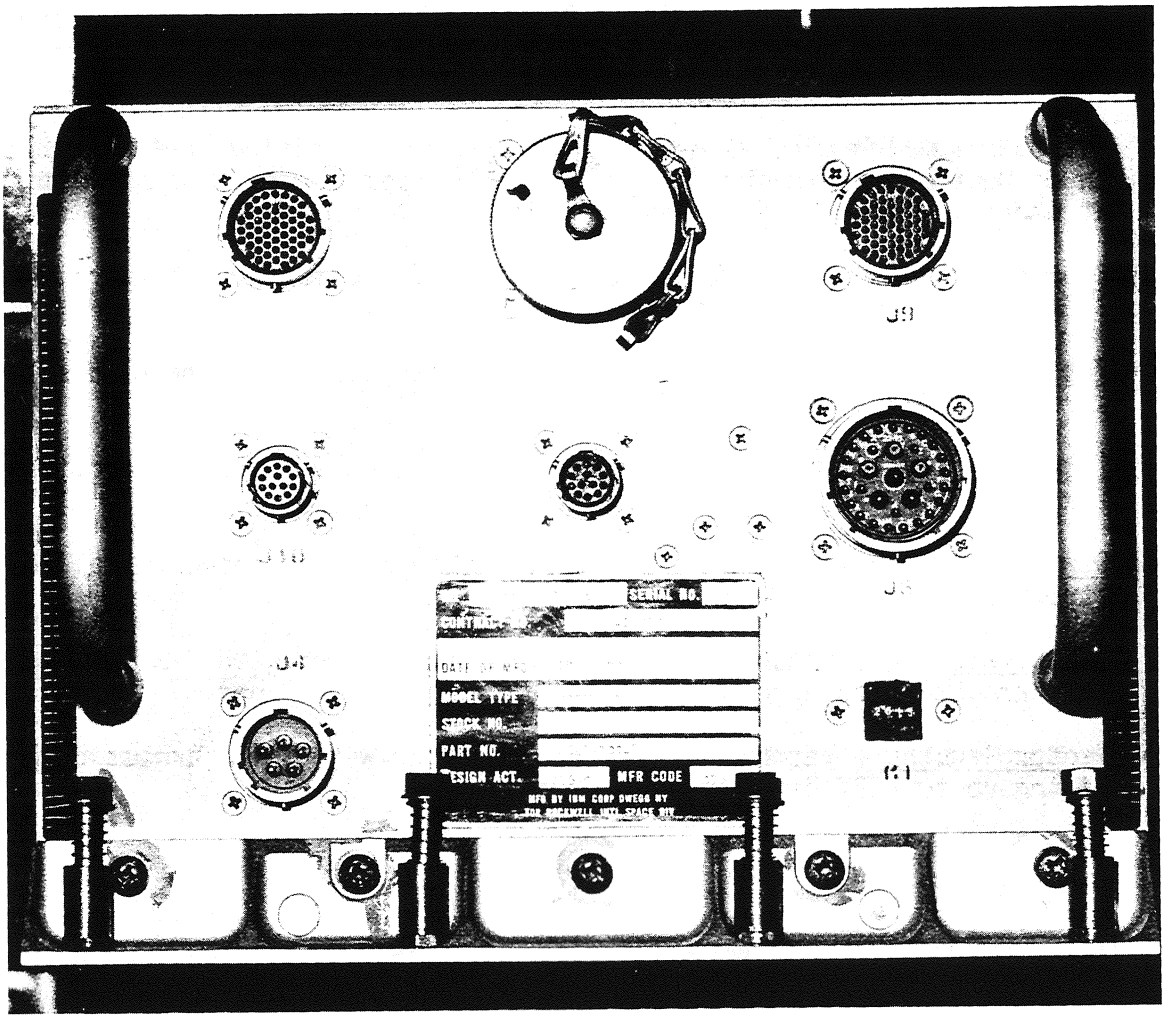
⑧ Position resolution - The resolution of characters on the CRT display screen is 1024 by 731.

⑨ Intensity levels - There are two intensity levels (normal and bright) for symbols.

⑩ Stroke structure - Characters are generated by line segments called strokes.

⑪ Flash rate - The flash rate for characters is 1 Hz with 5/8 second "on" time and 3/8 second "off" time.

⑫ Vector lengths - Vectors may be constructed with their largest component having up to 1024 different lengths.



Front view

Figure 4-18.- Display electronics unit.

4.4.3 DEU Physical Layout

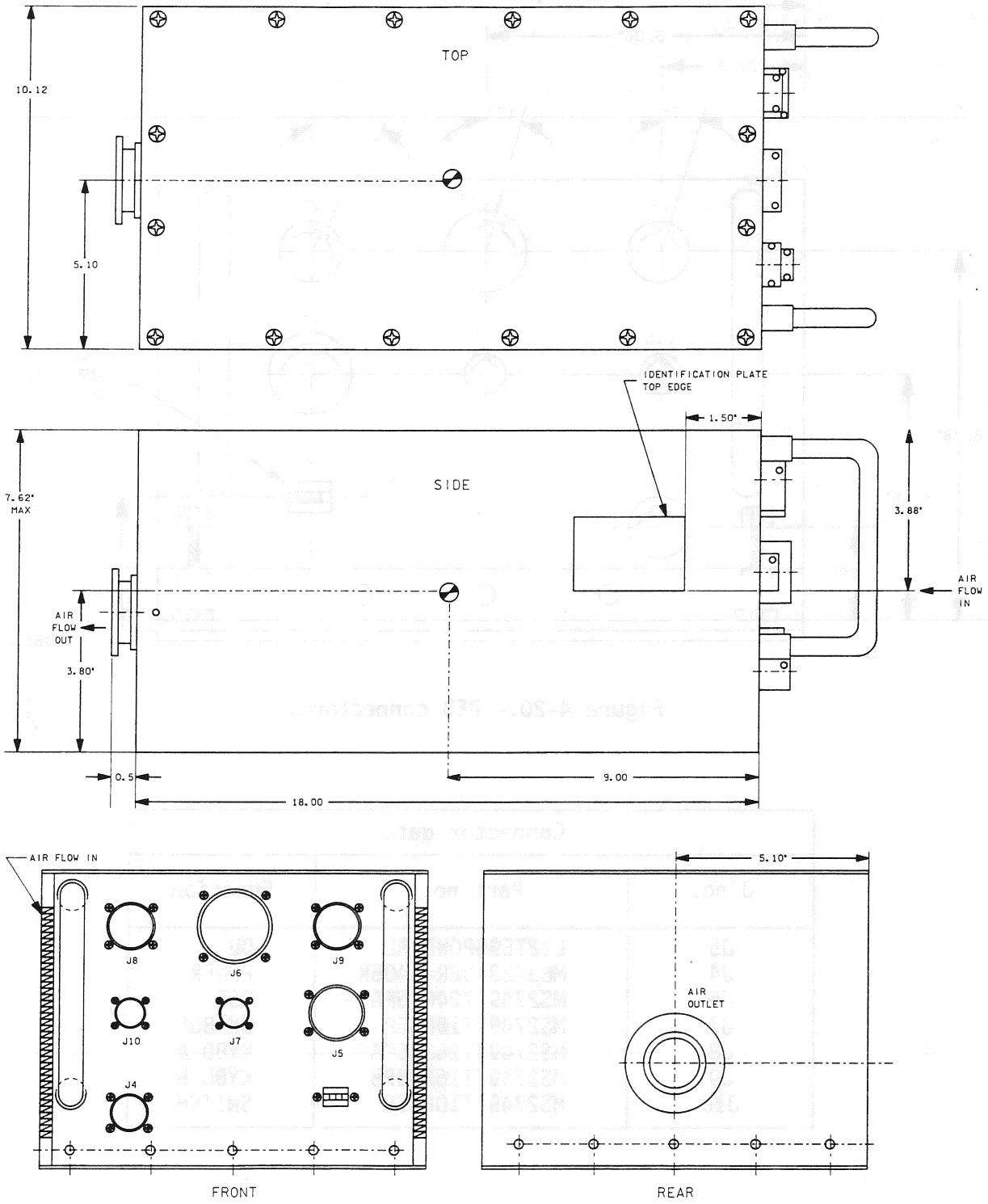


Figure 4-19.- DEU layout.

4.4.4 DEU Connectors

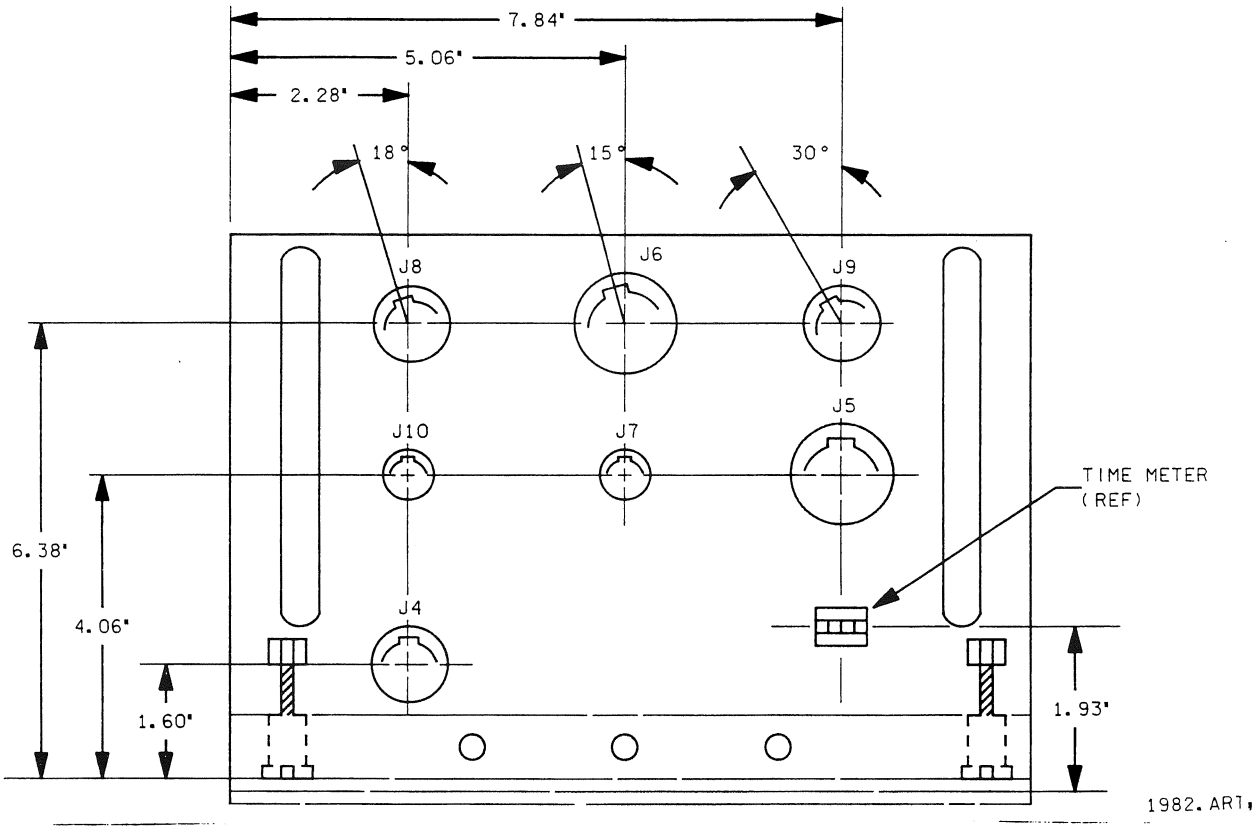
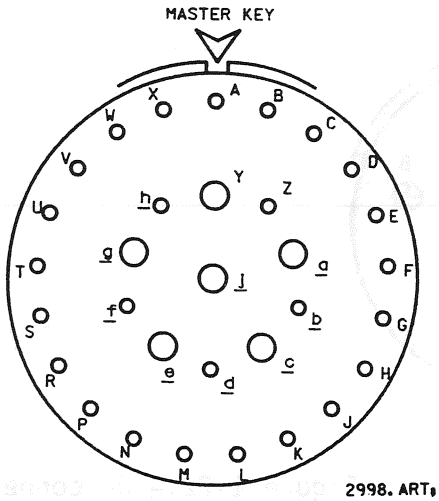


Figure 4-20.- DEU connectors.

Connector data		
J no.	Part no.	Function
J5	L22TE95PONA-AL	DU
J4	M83723-02R-1405N	POWER
J6	MS2749T724F35PA	GSE
J7	MS27497T10F35P	DK BUS
J8	MS27497T16F35PA	KYBD A
J9	MS27497T16F35PB	KYBD B
J10	MS27497T10F35S	SWITCH

J5 - DU

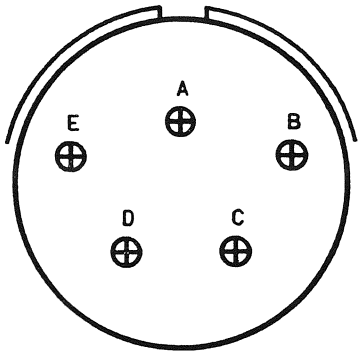


Pin	Function	Pin	Function
A	DEU chassis ground	T	Spare
B	DEU BITE signal	U	
C	DEU BITE RTN	V	
D	DEU BITE status	W	
E	DU deflection status	X	
F	DU video status	Y	Triax cable: X deflection, X return
G	DU phosphor prot stat *	Z	Spare
H	DU PWR supply stat	a	Triax cable: Z angle, Z return
J	DU CRT filament current	b	Spare
K	DU temp stat	c	
L	DEU/DU logic disc RTN	d	
M	Spare	e	
N	DEU/DU PWR on disc	f	
P	Spare	g	Triax cable: Y deflection, Y return
Q		h	Spare
R			
S			

*DU phosphor protect is not wired from DU in flight units.

Figure 4-21.- J5 connector.

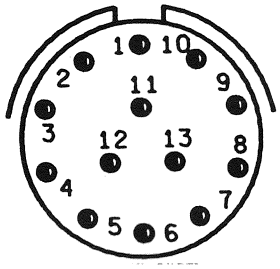
J4 - Power



Pin	Function
A, B	28 V dc PWR
C, D	28 V dc PWR RTN
E	DEU chassis ground

Figure 1-22.- J4 connector.

J 7 - DK Bus
J10 - Switch



2999. ART, 2

J 7

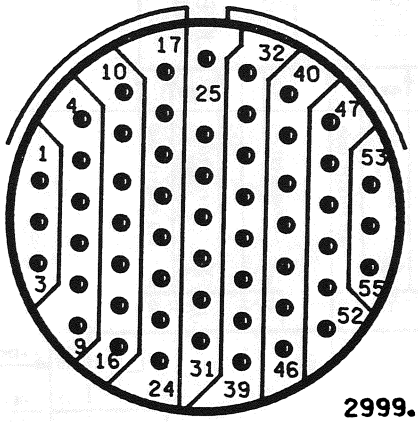
Pin	Function
1	Logic 1
3	Logic 0, ground
5	Bit 1 (MSB)
7	Bit 2
9	Bit 5 (LSB)
2, 4, 7, 8	Spares
11	Data bus A
12	Data bus B
13	DEU chassis ground

J 10

Pin	Function
1	DEU load
2	DEU load RTN
11	DEU load not
3	Mode 1, GNC
4	Mode 2, SM
5	Mode 3, PL
6	Mode RTN
7	KYBD unit SEL 1 (KYBD adapter A)
8	KYBD unit SEL 2 (KYBD adapter B)
9	KYBD unit SEL RTN
10	Remote PWR CNTL
12	DEU signal RTN
13	DEU chassis ground

Figure 4-23.- J7 and J10 connectors.

J8 - KYBD A
J9 - KYBD B



2999. ART, 2

J 8

Pin	Function
1 through 32	32 switch lines #
33	GRND A
34	GRND A
35 through 54	Spare (not used)
55	DEU chassis GRND

J 9

Pin	Function
1 through 32	32 switch lines #
33	GRND B
34	GRND B
35 through 54	Spare (not used)
55	DEU chassis GRND

*Detailed pin-to-switch layout is available in section 1.2.3.1, KYBD connector pin layout.

Figure 4-24.- J8 and J9 connectors.

4.4.4.1 DEU Connector Pin Layout

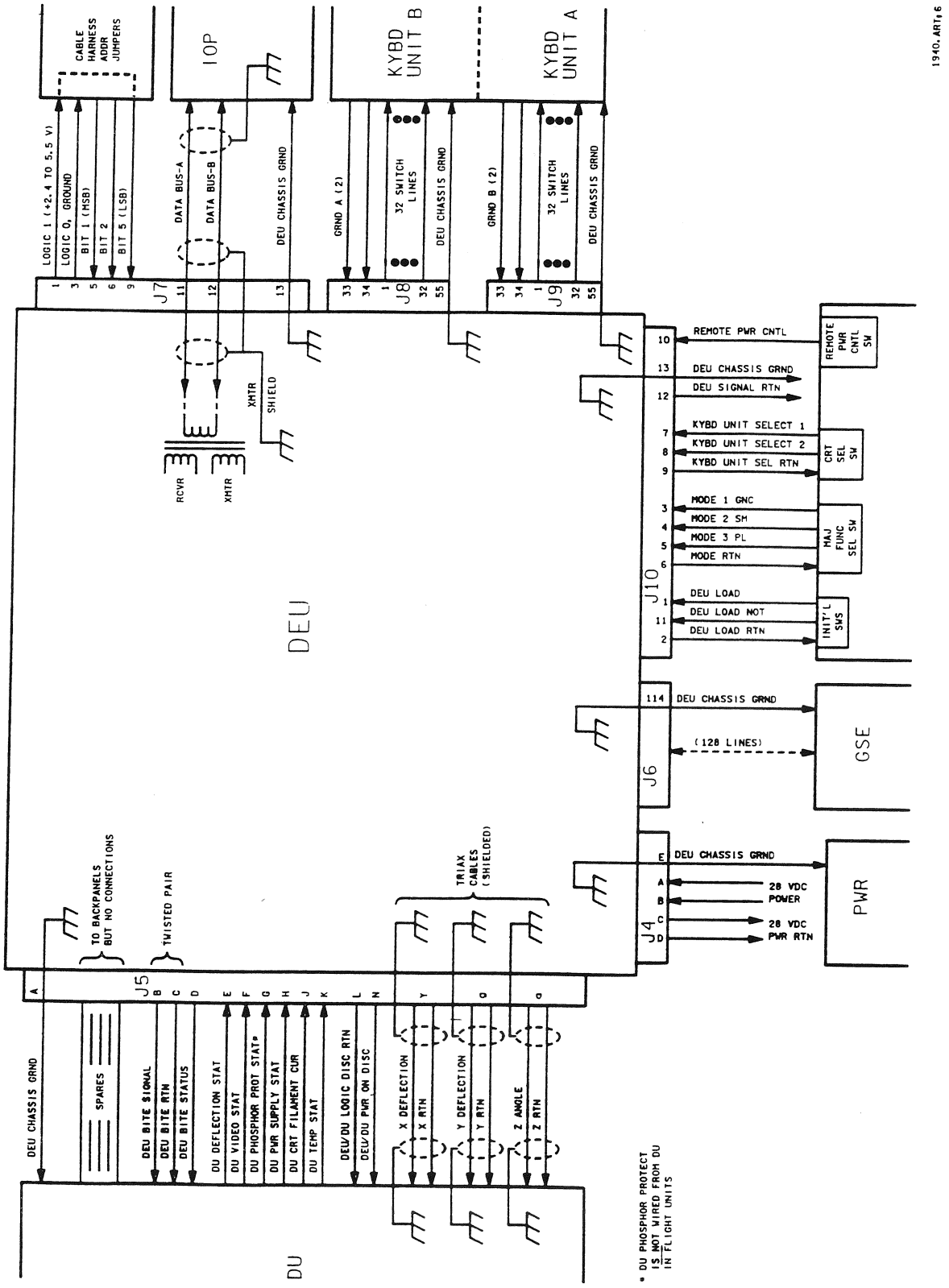


Figure 4-24.- DEU connector pin layout.

4.4.5 Power Supply

The DEU contains a power supply unit which provides ground isolation, voltage inversion, voltage scaling, and voltage regulation in order to efficiently supply steady voltages to DEU circuits and devices.

The DEU power supply also provides a "DU power on" discrete which, at the time of DEU activation, allows the display unit to be powered on simultaneously. (At DEU POWER ON, the DU also receives power; however, it cannot proceed to utilize that power for anything other than filament warming until it receives the power on discrete from the DEU.)

Each DEU/DU combination has a unique power control switch label "CRT X power" which, when turned ON, allows Orbiter main bus power (28 V dc) to enter each (DEU and DU) of the power supply units. The switch has three positions: ON, STANDBY, and OFF. The STANDBY position allows voltage to enter the power supply of both the DEU and its DU. Although the STANDBY position allows power to DU low voltage supply (for CRT filament warm-up), it does not provide the DU with the "power on discrete". Therefore, only the ON position allows full DEU/DU operation for CRT display. Power is routed through Remote Power Controllers ((RPC's), 10 amps) which are controlled via Orbiter control bus power.

	Orbiter power		TLM - MSID	
DEU/CRT	CNTL bus	MN bus	ON	STBY
1	AB1	MNA FPC 1	V73S2001E	V73S2002E
2	BC2	MNB FPC 2	V73S2011E	V73S2012E
3	CA1	MNC FPC 3	V73S2021E	V73S2022E
4	CA2	MNC FPC 3	V73S2051E	V73S2052E

Power consumption

A DEU consumes: 202 W - ON
0 W - STBY, OFF

BITE

A DEU "power supply error" will result in a DEU no-go condition, and subsequently trip the DEU mechanical BITE flag indicator located on the CRT faceplate. NOTE: A DEU PWR SUP error is not set in any of the DEU's hardware or software BITE registers. Therefore, there is no downlisted information for this condition. Besides the mechanical BITE flag indication, the DEU may exhibit characteristics of a powered off DEU (i.e., CRT blank, Input/Output (I/O) error CRT, and Initial Time Out (T/O) by common set GPC's). On the other hand, the failure may be such that only one particular type of voltage has been detected as inadequate and failed. In such cases, the DEU may appear to run normally (i.e., process keyboard inputs, update displays, etc.). The DEU should be interrogated carefully as some part of the power supply may have failed.

4.4.5.1 Power Supply Drawing

4.4.6 DEU/KYBD Interface

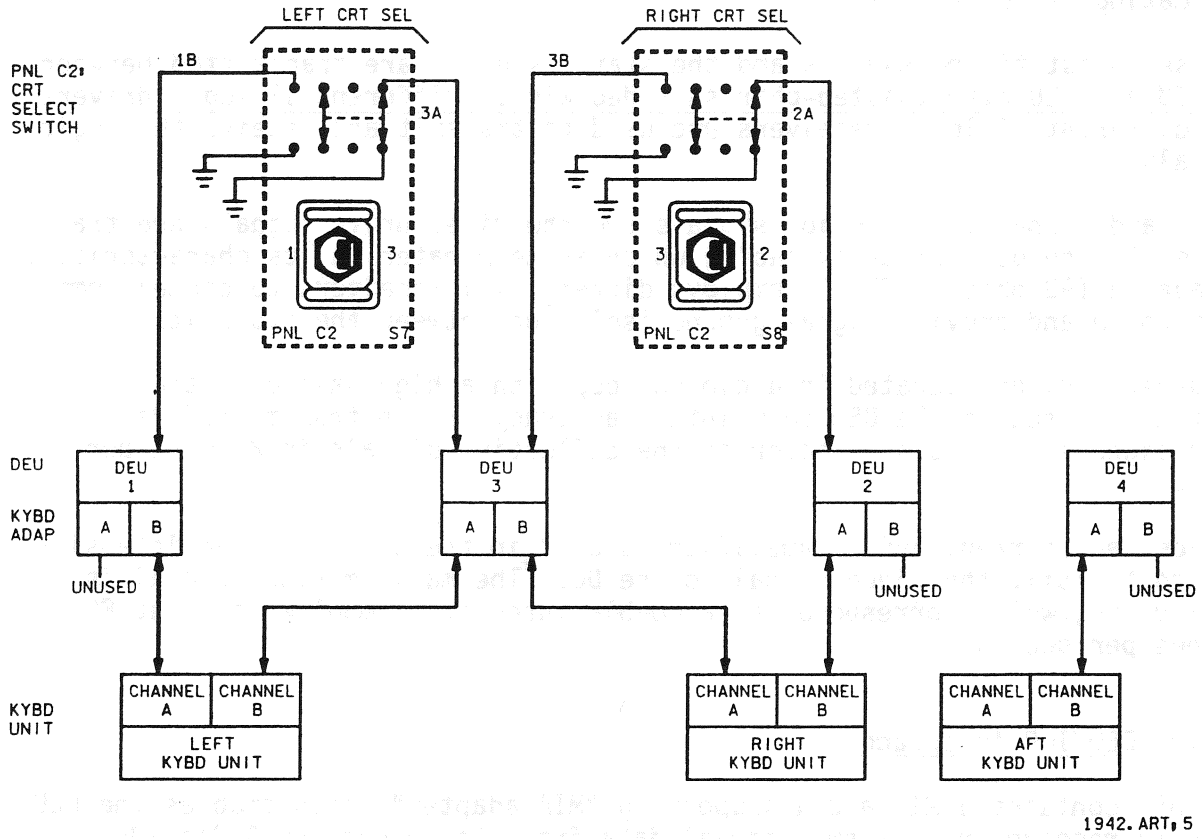


Figure 4-27.- DEU/keyboard interface.

In the forward cockpit, there are two KYBD units and three DEUs. Assignments are controlled by two CRT select switches. Each keyboard has access to two DEUs/CRTs via two contact sets or microswitches, referred to as channels "A" and "B". These keyboard channels are hardwired to the keyboard adapters in the DEUs (unfortunately also referred to as "A" and "B"). The keyboard adapter inside the DEU is activated via the CRT select switches.

4.4.7 DEU/DU Interface

The DEU provides the DU (CRT) with X and Y deflection, video, DEU BITE status, and self-test timing signals. The DU returns status signals to the DEU indicating its BITE status.

The self-test timing signals and the status signals are transmitted between the DU and DEU over twisted-pair shielded wire. Differential logic drivers and differential logic receivers are used to transmit and receive the logic signals.

The X and Y analog deflection signals and the video drive signals are transmitted to the DU over triax cable, which is terminated in its characteristic impedance (93 ohms). The DU employs differential receivers to cancel common mode noise and provide signal ground isolation between the two units.

A current driver operated in a closed loop with a high-gain operational amplifier is used in the DEU to maintain accuracy in the transmitted deflection signals. The scale factor of the deflection signals is 2 volts per inch.

A wide-band current buffer amplifier is used in the output of the intensity D/A to transmit the video signal to the DU. The maximum video signal is 5.33 volts, which corresponds to a double intensity symbol written at 80,000 inches per second.

4.4.8 DEU/IOP Interface

The DEU contains a MIA and a supporting "MIA adapter" which enables the DEU to both receive and transmit serial data from its commanding GPC's IOP.

4.4.8.1 MIA

4.4.8.1.1 Receiving data. - Serial data are received from the GPC by a standard serial MIA. The MIA accepts valid commands and data words from the serial data bus and converts the received data from Manchester format to Non-Return-to-Zero (NRZ) format for serial transfer to the MIA adapter. The MIA also determines the validity of the received data by checking for parity, bit count, and nonvalid Manchester errors.

Once serial data are received by the MIA, they are then transferred to the MIA adapter where they are converted to parallel form and buffered in registers prior to transfer over a 16-bit parallel bus to the "display controller". Buffering is required because of a timing difference between the GPC bus and the small processor in the DEU. The MIA adapter signals the small processor in the DEU that it has commands or data words for transfer by issuing break-in requests that cause the small processor to store the data in memory. If the command word is a request to receive data, the word count contained in the command word is stored in the MIA adapter. This count is

decremented each time a word is stored. When it equals zero, a last-word-stored interrupt is generated by the MIA adapter, which terminates the data transfer.

4.4.8.1.2 Transmitting data. - When a GPC-to-DEU command to "transmit data" (i.e., MCDS status request or poll) is received by the DEU's small processor, software DEU Control Program (DCP) sets up an output buffer of data to be transmitted. The word count of this transmission is stored in the MIA adapter. The MIA adapter issues break-in requests to the DCP that cause the small processor to access memory for the data to be transmitted. The data are then transferred over a 16-bit parallel output bus to the MIA adapter, where they are converted to serial form via the adapters' buffer registers. The word counter, which knows the word length of the entire transmission, decrements the count by one for each word transmitted. When it equals zero, an interrupt signals the DCP to terminate the transfer of data. As the data leave buffer register 3, they are in Manchester serial format and are shipped to the MIA. The MIA verifies the validity of the NRZ-to-Manchester format, generates a data sync word, and adds parity for transmission to the IOP.

4.4.8.2 MIA adapter

The MIA adapter consists of the following registers and control gates which it utilizes to interface MIA I/O's to/from the small processor in the DEU controller.

Register 1 and controls receive 25 bits of serial NRZ data from the MIA receiver or 16 bits of parallel data from the small processor over the output bus for transmission by the MIA. When MIA inputs are received, this register outputs 19 data bits to register 2 for buffering, the 5 bits containing the DEU address are output to the address compare circuits for valid address test, and the 3 SEV bits are sent to error check and control to verify the proper 101 pattern. When data are transmitted, 16 bits are received on the parallel data-out bus from the small processor.

Register 2 and controls are used to buffer the 19 data bits received from register 1 for MIA-received data, provided that the address compare was valid, or 16 data bits from register 1 for transmission by the MIA transmitter. This register parallel-outputs 16 data bits to register 3. Nine bits of a valid addressed receive command word are sent to the word counter/multiplexer provided that parity, bit count, invalid Manchester, and echo check circuits indicate error-free data.

Register 3 and controls constitute a 24-bit register used to serialize, for MIA transmission, the 16 bits received from the small processor, 5 MIA address bits, and 3 SEV bits. Data received from the MIA are input to the small processor from this register via the 16-bit parallel input bus.

The word counter/multiplexer is used to count the number of words received or transmitted. When used for input data (receive), it is loaded with a nine-bit word count field contained in the input command word. When data

are transmitted, the word counter is loaded from the small processor via the data-out bus. The word counter is decremented for each word received or transmitted, and provides break-in and interrupt control to the data flow.

Error check control is used to set the hardware status register and/or the DEU BITE indicator for parity, bit count, invalid Manchester, and echo check failures. The wrap register 1 control section converts the serial first word transmitted by the MIA, when in the programmed BITE mode of operation, to parallel form and makes it available to the small processor on the 16-bit parallel input bus.

Data flow interface - The small processor data flow interface provides the break-in and interrupt control for interfacing the MIA data, both received and transmitted, to the small processor.

4.5 HIP POCKET IFM PROCEDURES

CRT 2 Switches IFM:

If any of the following switches fail for CRT 2:

- MF SWITCH
- RT KYBD SEL
- DEU LOAD
- Partial PWR switch (Contact for RUN DISCRETE only - must have contact for RPC - no visibility, but this procedure will do check - will not work for BC 2 bus fail)

A. Power down CRTs 2 and 4

B. Remove panels for CRTs 2/4 - they are sitting side by side on Pilot's side of spacecraft. Use IFM steps for DEU changeout

C. Disconnect the J10 connector from DEU 2 and DEU 4. Install DEU 4 J10 connector to DEU 2. I believe it will reach that far. Tape DEU 2 J10 connector out of the way.

D. Power up CRT 2 and CRT 4. The following are the impacts to procedure:

   - CRT 4 will not be functional (no PWR ON discrete)
   - CRT 4 MF and LOAD switch will now work for CRT 2
   - CRT 2 MF and LOAD switch will not be functional
   - Right KYBD will be active on CRT 2 all the time regardless of position of RT KYBD SEL switch. Caution: if RT KYBD SEL switch functional and in "3" position, keystrokes on RT KYBD will go to both CRTs 2 and 3. Use LEFT KYBD for CRT 3 keystroking.

Prior to committing to a DEU IFM, this power check should be used to verify power being fed to the DEU is failed; i.e., not a switch failure. Remember, only one set of contacts provides control power to the RPCs; TLM is on the other set.

IFM to check power to DEU X:

A. To obtain access to DEU X, remove panels L16 and L8 (5/32 Allen-head screws)

B. Panel C2 - CRT #X, PWR - OFF

C. At DEU X, remove connector from J4 (ref. photo on page D-17 of IFM checklist)

D. Use multimeter, set for dc volts, 200V scale; attach red lead to pin A of cable connector and black lead to pin C of cable connector.

E. Panel C2 - CRT #X PWR - ON

F. Check for indicated +28V dc on multimeter.

G. If voltage is present, DPS should advise changeout of DEU or CRT.

H. If no voltage, CRT #X is no-go; CDR can move to RH seat or let PLT make the landing.

DEU DISCRETE IFM (MF Switch, Select Switch, DEU LOAD):

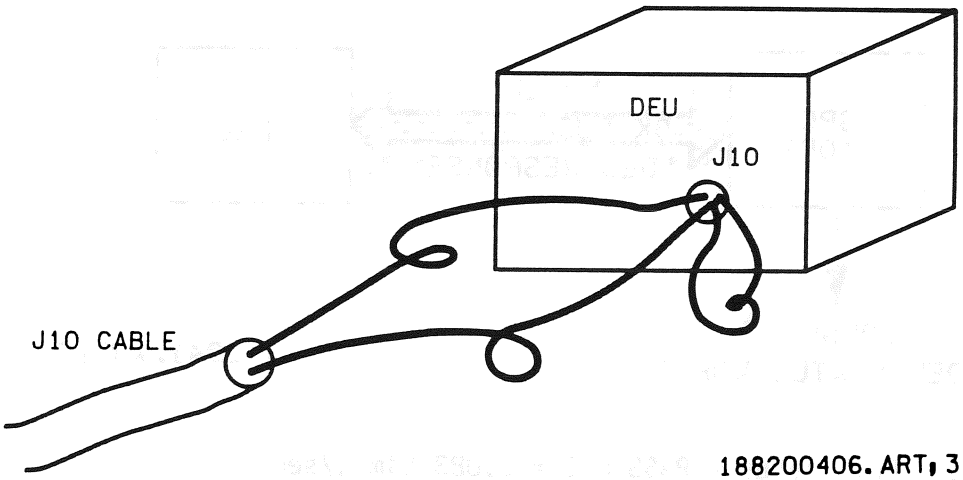


Figure 4-28.- DEU Discrete Fail IFM.

Example: For DEU 1 select switch fail.

A. Jump from cable to DEU the following pins:

3 to 6 (MF switch)
10 and 12 (DEU power ON)

Note: Probably do not need pins 1, 2, and 11 for DEU load.

B. At DEU, jump pin 8 (Select 2) to pin 9 (Return)

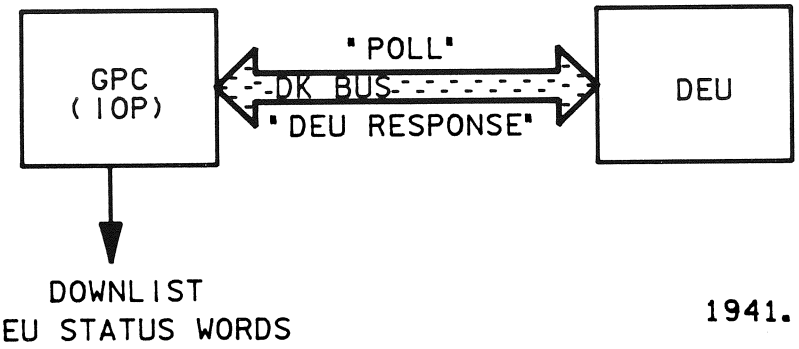
CRT 1 and CRT 3 cable swap IFM:

Although not normally considered, CRT 3 cables can be moved to CRT 1. This may be considered if CRT 1 is failed due to a bus problem and CRT 3 is failed at the DU or DEU. It should be noted that, although CRT 3 cables can reach DEU/DU 1, CRT 1 cables cannot reach DEU/DU 3. Therefore, this IFM can only work one way.

4.6 MCDS OPERATIONS

4.6.1 GPC/MCDS Data Communications

Communications between the GPC and the MCDS are conducted on a demand-response basis in which the IOP commands the DEU to receive or transfer data. IOP-to-DEU commands include display updates, time data, and request for DEU status (commonly referred to as a "poll"). DEU-to-IOP communications or "the answer" are called either the "DEU response" or "poll response."



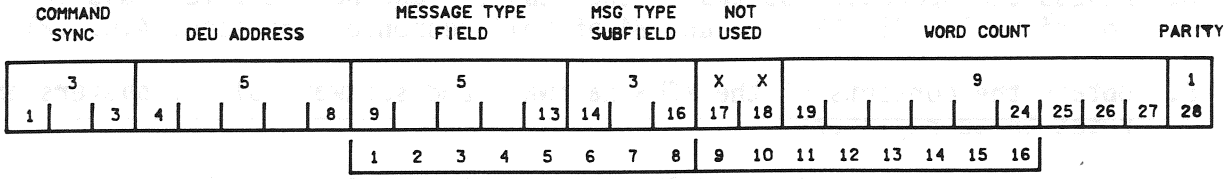
1941.ART;3

NOTES: (1) Poll rates - PASS GPC = 2.083 times/sec
BFS GPC = 3.0 times/sec

(2) "Poll" consists of a channel time fill command and a 1 word status request, where the "poll response" usually consists of 16 words (word 16, a checksum word is not downlisted from the PASS GPCS).

4.6.1.1 GPC POLL Detail

The following command word is used for IOP-to-DEU communication:



1937.Art 2

The DEU memory provides storage for 16-bit full words. If a command word is stored in DEU memory, bits 9 through 24 of the "poll" are stored, with all other bits checked but not stored.

Bit position

1-3 Command sync. IOP identifies command word as a "poll."

4-8 DEU address. Interface Unit Address (IUA).

9-13 14-16 Message-type fields. These two fields contain a five-bit and a three-bit code that together define the specific message being commanded. The codes are:

MSG no.	Field	Subfield	Title
1	00000	100	MCDS STATUS REQUEST
2	00010	XXX	IPL BITE STATUS REQUEST
3	00100	XXX	RESET SCRATCH PAD LINE
4	00110	XXX	BUFFER FILL
5	11100	000	TIME FILL
6	11100	011	DISPLAY FILL
7	11100	101	FORMAT FILL
8	11101	XXX	DEU MEMORY DUMP

17-18 Not used

19-27 Word count. Nine-bit code that indicates the number of command data words to follow the command word. If the word count = 0, the command word is the only message transfer. If the word count ≠ 0, the number indicated is the count of command data words to follow the command word.

4.6.1.1.1 MSG 1 - MCDS STATUS REQUEST. - The MCDS status request message is utilized by the GPC to perform the following functions:

A. Assess the operating status of the MCDS by reading the five flags in bit positions 20, 21, 22, 23, and 24 of poll response word (sec. 4.6.1.2).

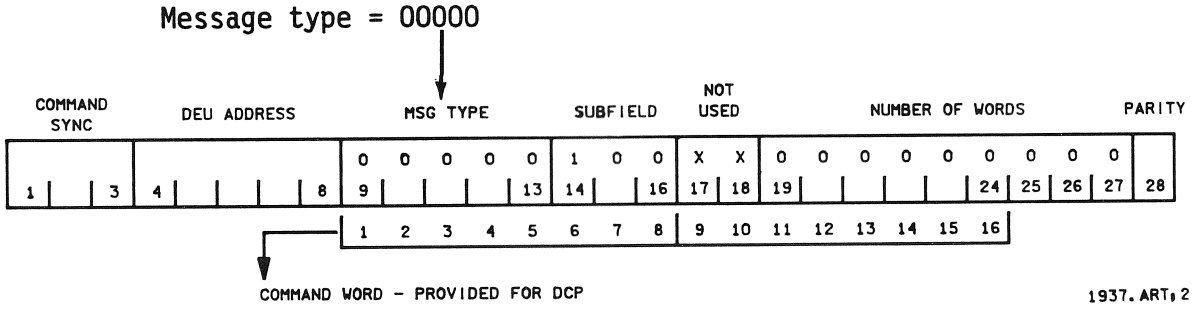
B. Obtain the contents of the MCDS hardware and software BITE registers and reset these registers.

C. Transfer up to a maximum of 30 key code strokes from the MCDS to the GPC.

D. Obtain a 2's complement checksum word to verify proper transfer of information during the response.

The MCDS status request is valid for either the DCP or Initial Program Load (IPL) software. The response from each is different. The DCP returns 16 response data words (see next page) while the IPL software returns only 1 data word. The IPL response is covered in detail in the IPL section of this document.

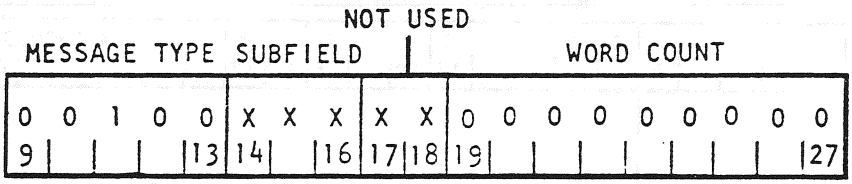
MCDS Status Request



This request is "answered" by the DEU via a typical MCDS status response, detailed in section 4.6.1.2, POLL RESPONSE DETAIL.

4.6.1.1.2 MSG 2 - IPL BITE STATUS REQUEST.- This GPC poll command is covered in detail in the IPL section of this document.

4.6.1.1.3 MSG 3 - RESET SPL.- The reset scratch pad line message is used by the GPC to reset the displayed keystrokes in the scratch pad line.

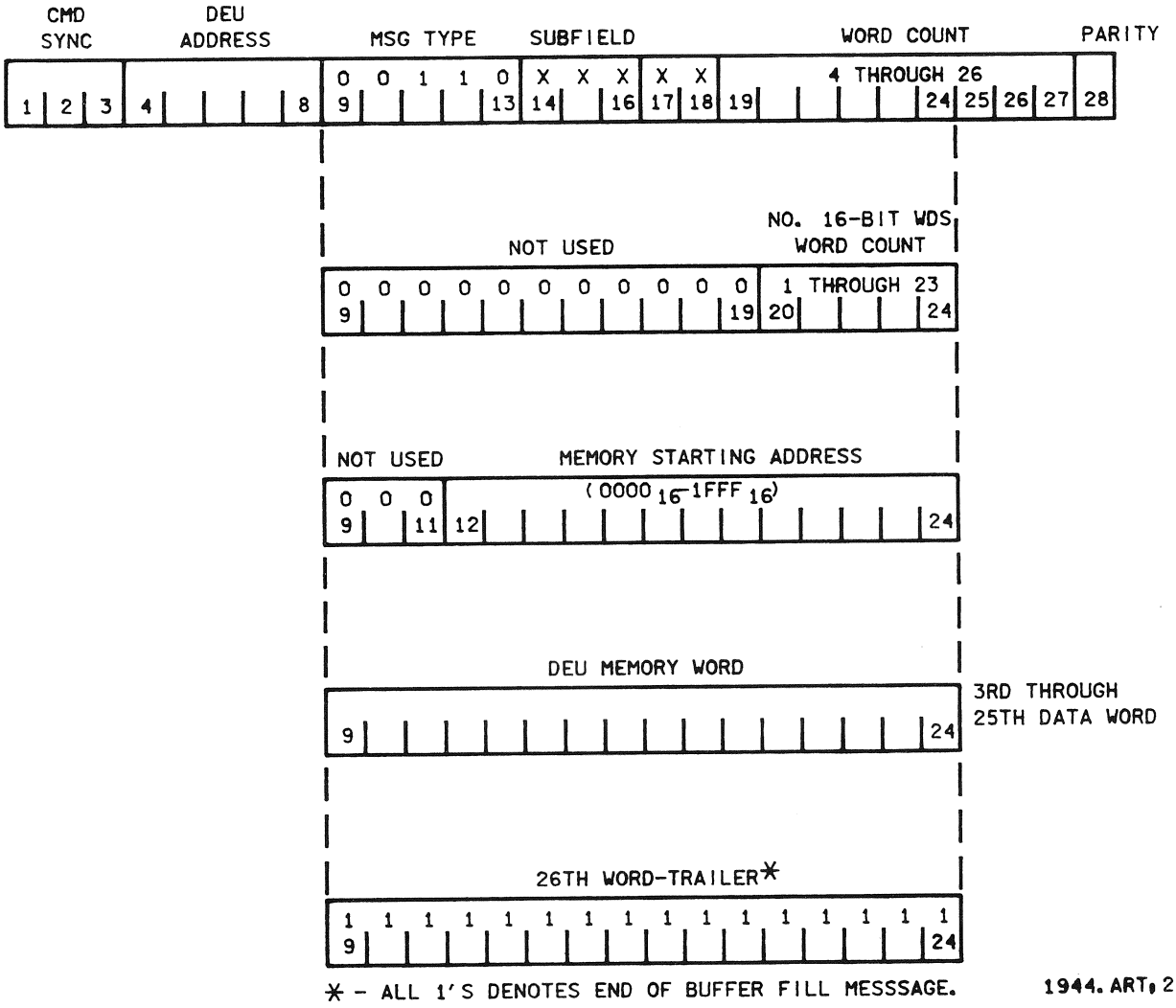


Bit position

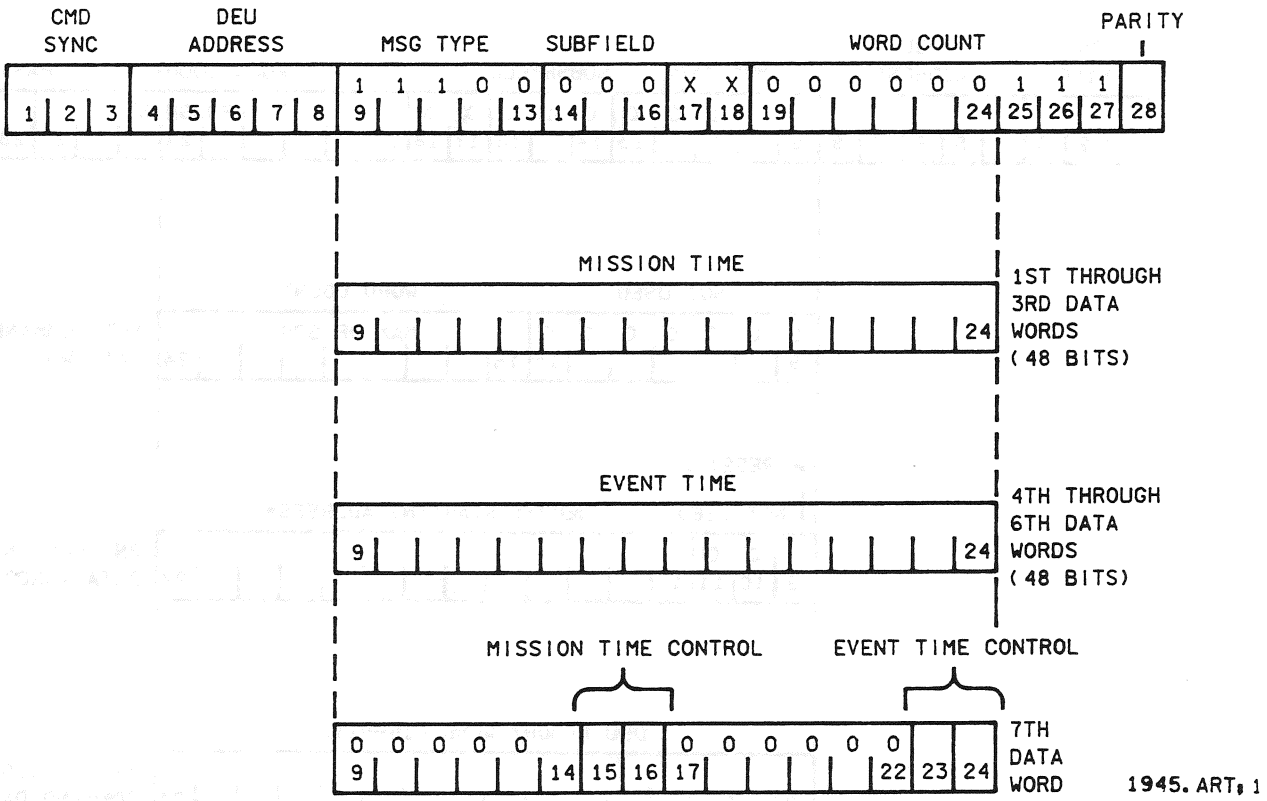
9-13 14-16 Message type fields: 00100 XXX indicates a command to reset the SPL

19-27 Word count: All 0's indicates that no command data words follow.

4.6.1.1.4 MSG 4 - BUFFER FILL.- The buffered fill message is used to store data into the DEU memory. It requires full buffering; i.e., the entire message is stored into a buffer and then transferred to the specified storage addresses. It is therefore limited to 23 16-bit memory words and used primarily for test purposes.



4.6.1.1.5 MSG 5 - TIME FILL.- The time fill message is used by the GPC to transmit mission and elapsed time data to the DEU in conjunction with polling. Both time values are 48-bit floating point numbers with a unit value of seconds. The time fill message also contains control information to allow starting, stopping, incrementing, or decrementing the time values. The time fill message is issued immediately prior to the status request message that is used to poll the MCDS. The message sequence for time fill messages consists of a command word to identify the message and seven command data words. The unique contents of the message words are as follows:

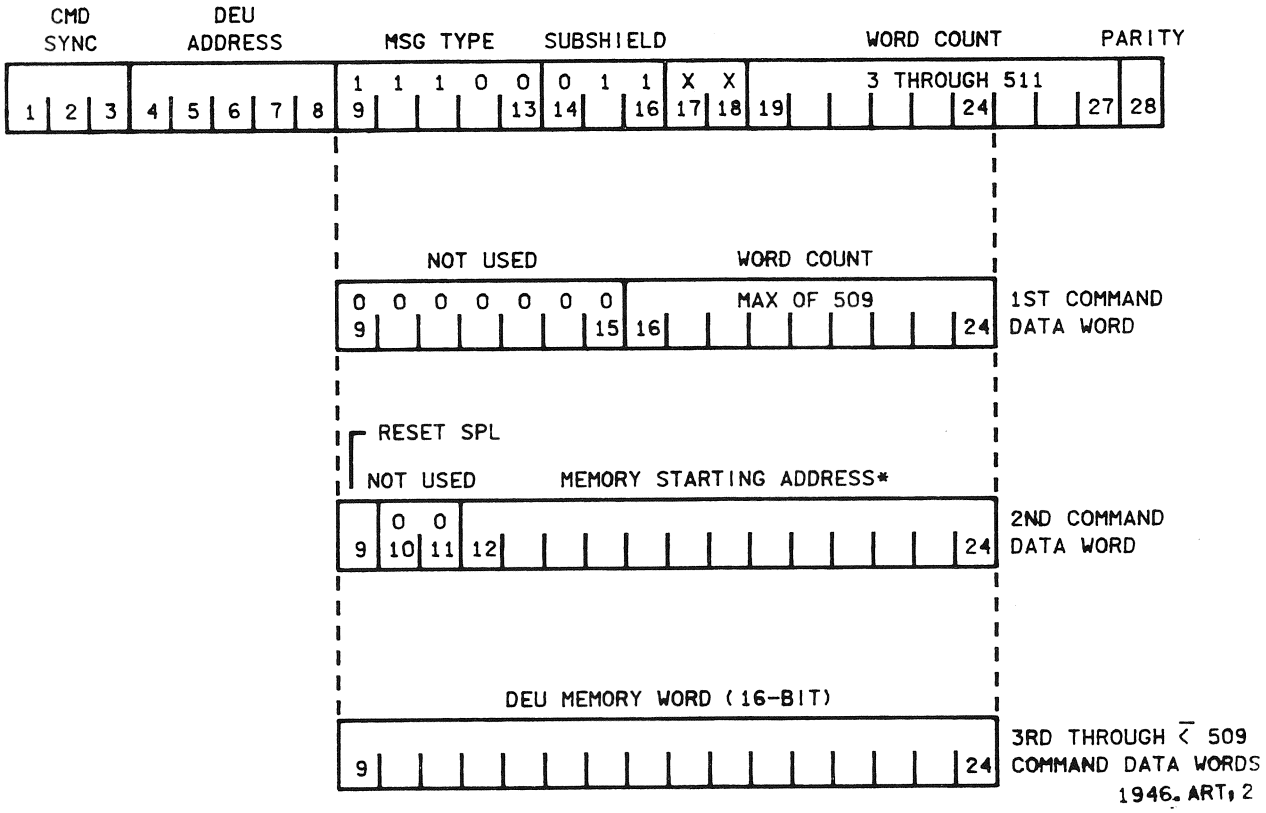


Bit position

15	A 0 increments and 1 decrements the mission time.
16	A 0 starts and a 1 stops the mission time.
23	A 0 increments and a 1 decrements the event time.
24	A 0 starts and a 1 stops the event time.

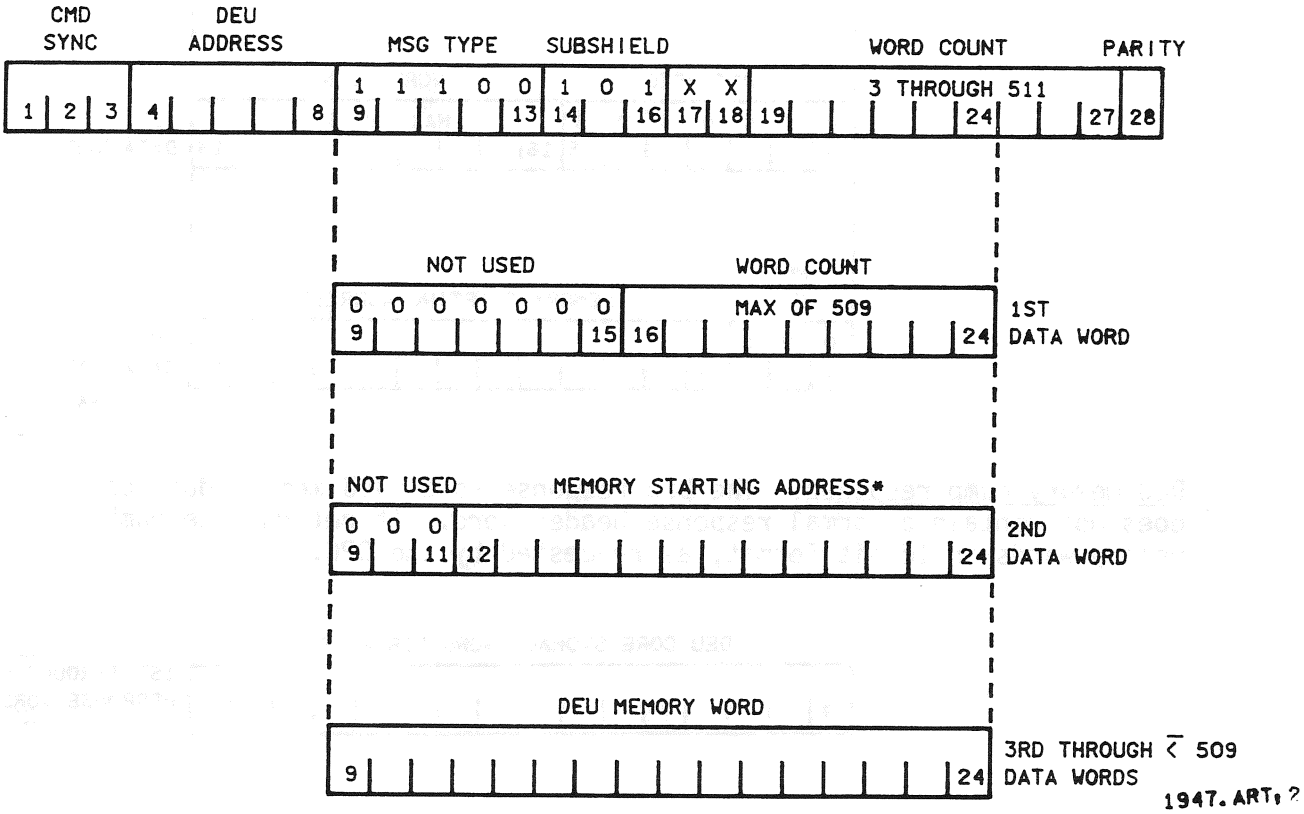
4.6.1.1.6 MSG 6 - DISPLAY DATA FILL.- The display data fill message is used by the GPC to load Format Control Words (FCW's) directly into the data display buffer or the message line buffer areas of the DEU memory. The FCW's are used by the symbol generator to generate the displays; the types of FCW's are described later in this section under Refresh Control. During IPL, the display data fill message may be used to load data to any location of memory; but during DCP operation, it is limited to the two areas specified.

The message sequence for display data fill consists of a command word to identify the message followed by the memory load data as command data words. The unique contents of the message words are as follows:



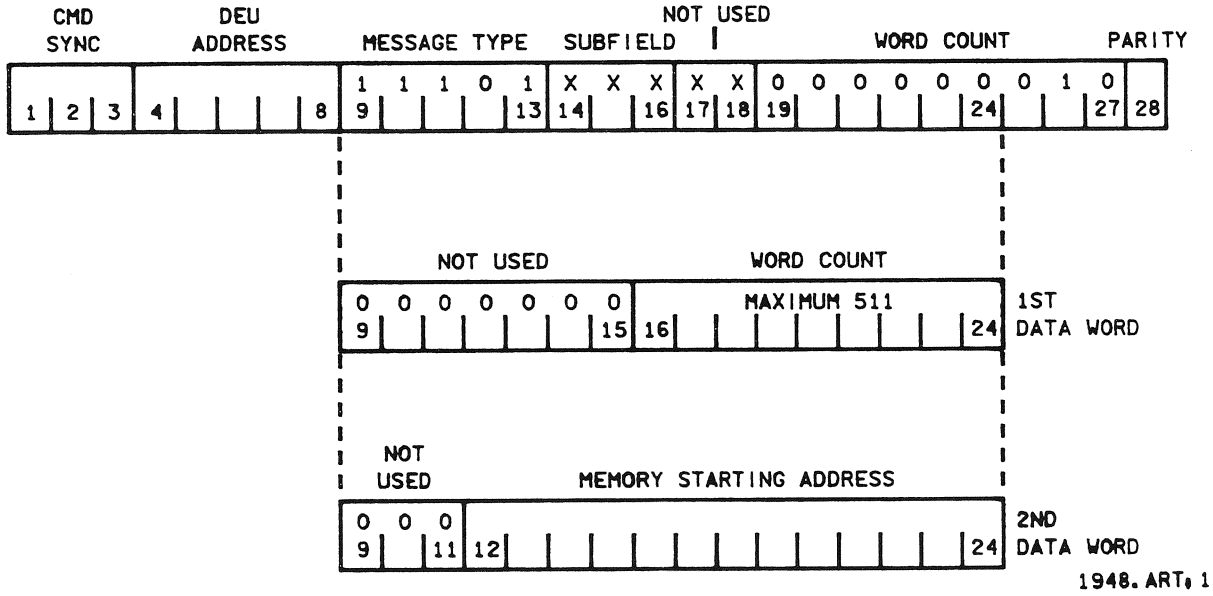
*Memory starting address. A 13-bit field that specifies the starting address within the DEU memory for the following data words. During IPL, this start address may be anywhere in the DEU memory. If not during IPL, the start address must be within the DCP-specified GPC data display buffer or message line buffer. In addition, the next (third) data word must be delayed a minimum of 587 μs to ensure sufficient time is available for the software to modify addressing. A pointer address value is set, the starting address specified, and the 16-bit DEU memory words are stored directly into specific sequential addresses.

4.6.1.1.7 MSG 7 - FORMAT DATA FILL.- The format data fill message is used to load data into the DCP specified critical format buffer area of the DEU memory. If the last word in the format buffer is filled by this message, the DCP will checksum the contents of the format buffer, except for the last word; the calculated checksum will then be compared against the last word in the format buffer. A miscompare will cause the checksum bit in the DEU software status register to be set

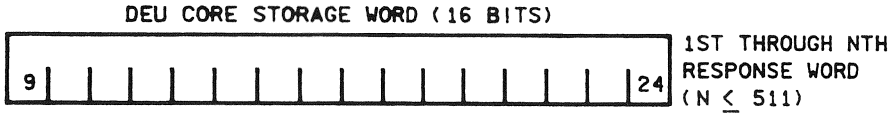


*Memory starting address. A 13-bit field that specifies the starting address within the DEU memory for the following data words. This start address must be within the DCP-specified critical format buffer area. In addition, the next (third) data word must be delayed a minimum of 568 μs to ensure sufficient time for software to change address pointer values. This allows storing of critical format data directly without buffering.

4.6.1.1.8 MSG 8 - DEU MEMORY DUMP.- The DEU memory dump message is used by the GPC to read the contents of DEU memory in GNC OPS 9 only.

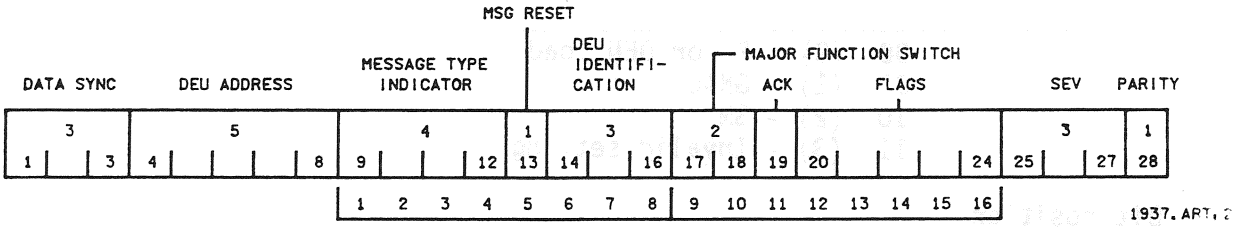


DEU memory dump response. The DEU response to a "DEU memory dump command" does not contain a normal response header word. It returns the number of memory words in 16-bit format, as requested by the GPC.



4.6.1.2 POLL RESPONSE Detail

The format of DEU-to-IOP poll response consists of a standard header word, word 1 defining the response, followed by the required amount of detailed supporting information. The poll response header word is:



The DEU memory provides storage for 16-bit words. The detailed information of the response, bits 9 through 24, is generated in memory, with bits 1 to 8 and 25 to 28 added before transmission to the IOP.

Bit position

1-3 Data sync. Identifies word as "poll response."

4-8 DEU address. Interface Unit Address (IUA).

9-12 Message-type indicator. Four-bit code that defines the specific type of response to the GPC being provided by this or the following response data words. The codes are:

MSG no.	Bit indication	Title
1	0000	MCDS STATUS RESPONSE
2	0011	IPL STATUS RESPONSE
3	0101	IPL BITE STATUS RESPONSE

13 MSG reset. A logic one indicates message reset.

14-16 DEU identification. Three-bit code that identifies the responding DEU. This ID code is supplied by the GPC at DEU initialization, or reassignment by the GPC, resides in DEU memory, and is used by the GPC to identify the data.

17-18 Major function switch. Two-bit code that repeats the position of the MCDS major function switch on center console control panel C2 and is used to define the software function (payload management; guidance, navigation and control; or system management/performance monitor) of the response data so that all GPC's associated with the function can use the data.

00 (0) - PL or DEU load
01 (1) - GN&C
10 (2) - SM
11 (3) - Invalid setting

Bit position

19 ACK. A logic 1 indicates an ACK message.

20 Flag. Logic 1 indicates that display is in freeze status.

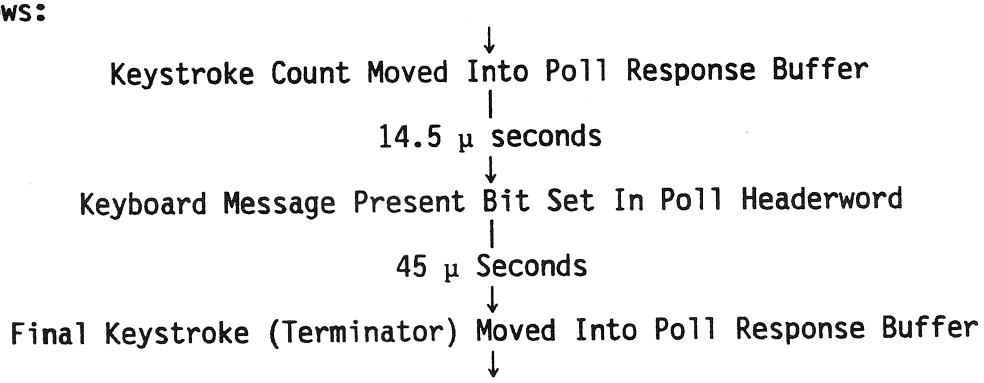
21 Flag. Logic 1 indicates that a complete keyboard message is present.

22 Flag. Logic 1 indicates stand alone self-test in progress.

23 Flag. Logic 1 indicates that a critical BITE has been detected.

24 Flag. Logic 1 indicates that initialization is required (DEU load).

4.6.1.2.1 DEU Control Program (DCP) timing windows that affect poll response.- During the DCP processing of terminator keys (PRO, EXEC, FAULT SUMM, SYS SUMM, RESUME), there are two timing windows that could permit erroneous transfer of the keyboard message to the GPC if polling occurs during the windows. These timing windows are a result of the specific placement of pertinent keystroke information into the poll response buffer as follows:



If a poll occurs in the 14.5μ second window, the keystroke count will be zeroed following the poll. On the next poll, the keyboard message present bit will be set in the headerword but the keystroke count will be zero. If a poll occurs in the 45μ second window, the poll response will be missing the final keystroke (i.e., the terminator). For example, the EXEC keystroke on ITEM entries will be missing. For single keystroke messages (RESUME, FAULT SUMM, SYS SUMM, EXEC, PRO) there will be no keystrokes in the poll response. Essentially, these conditions are caused by the poll response interrupting the keyboard handler. If polling occurs during one of these windows, one of four events may occur (depending on when the interrupt occurred):

A. An "ILLEGAL ENTRY" message for PASS, or an "I/O ERROR CRT X" message for the BFS is displayed on the CRT. This may occur due to the GPC's interpretation of a positive keystroke count > 0, but no associated keycode. If this occurs, press the MSG RESET key and then re-key the message.

B No indication of a response from the GPC. If no response from the GPC occurs during the nominal expected time, re-key the message.

C. A "POLL FAIL" message is displayed on the CRT. To rectify this situation, reassign that DEU to the controlling GPC from another keyboard. Key in GPC/CRT XY (where X is the previously controlling GPC and Y is the poll failed DEU), and then re-key the message.

D. A "Zero message length in DEU resp buf" GPC error (Group 06 Code 01) occurs. If this occurs, just re-key the message.

Note that the keystrokes on the CRT scratch pad line are unaffected by these timing windows and appear normal to the crew. The PASS, BFS, and ground processing of keyboard inputs often react differently to erroneous keystroke data. Table 4-I summarizes the effect of erroneous keystroke data on the different systems.

Previous to STS-9, the impact to pass processing of a data ITEM entry with a missing EXEC (terminator) was not fully realized. This specific problem could have resulted in erroneous data being placed in the ITEM numbers that the user intended to manipulate and/or data being placed in ITEM numbers that the user did not intend to manipulate. The PASS always annunciated an "ILLEGAL ENTRY" for this case even though the software has accepted the erroneous data. A program note was written in January 1983 (DR 52773) that somewhat documented this condition (this DR concerned DEU equivalents, but the causes were still the same). Later on in 1983 a source fix was implemented for OI-3 that prevented the software from accepting this erroneous data. Thus, the DR has since been closed.

For the long term, the timing window problems should be fixed at the DCP. Although the PASS will no longer accept erroneous ITEM type entries, there are still some potentially serious impacts from the zero message length timing window. For example, the BFS could respond to a DK listen-type keyboard entry when the PASS would reject the same entry. If this occurs on

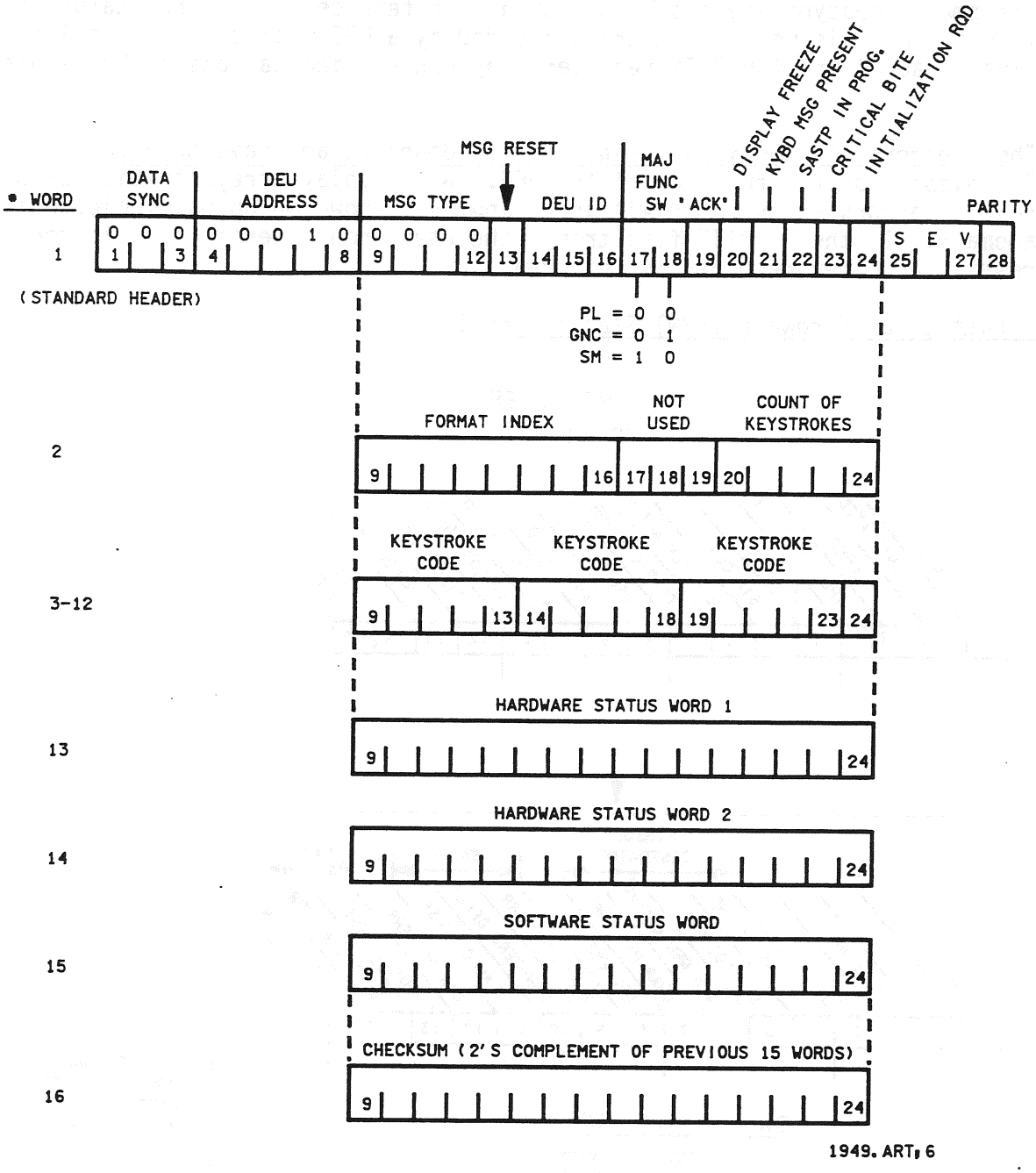
a GPC/CRT 5X EXEC entry, the BFS would take control of a CRT that the PASS does not give up resulting in dual commanders and a potential fail-to-sync. The current holdup in implementing a fix in the DCP is that the fix may require a recompile and reverification of the DCP. IBM had formulated a modification to the DEU memory that would fix this problem; however, RI/Dny stated that the DEU memory was already full, and that a scrub of the DEU code would have to be performed before a modification could be implemented. AT that time, no resources were available to pursue this option. Also, the assessment was made by RI/Dny that the probability to polling during these windows would be very small. Therefore, nothing was done and nothing is currently being done to rectify this issue.

TABLE 4.I.- DCP TIMING WINDOW KEYBOARD PROCESSING EFFECTS

Problem	Type keystroke	PASS effect	BFS effect	Ground effect
Zero message length (14.5 μ second window)	SPEC ABC PRO	Keyboard input lost. GPC error 0601 logged. No crew annunciation is made	Interpreted as SPEC A PRO. Only legal single digit SPEC is SPEC 0. Therefore, typically get ILLEGAL ENTRY	Keyboard input appears normal in MCC. Ground processing does not use keystroke count
	All others (1)	Same as above	Keystrokes processed normally. BFS only uses keystroke count for SPEC type entries	Same as above
	RESUME, FAULT SUMM, SYS SUMM, EXEC (i.e., single keystroke entry)	Keyboard input lost. "ILLEGAL ENTRY" annunciated	Keyboard input lost. "I/O ERROR CRT" annunciated	Complete keyboard input appears in the MCC on the next poll of the DEU
	ITEM A EXEC	Same as above	Same as above	Same as above
Missing terminator key (45 μ second window)	ITEM A + X EXEC (i.e., an ITEM entry with a single data item)	Interpreted as ITEM A + 0 + 0 + 0 .... until an "ILLEGAL ENTRY" condition is encountered (2)	Same as above	Same as above
	ITEM A + X + Y + Z EXEC (i.e. an ITEM entry with multiple data items)	Interpreted as ITEM A + X + Y + 0 + X + Y + 0 + X + Y + 0 .... until an "ILLEGAL ENTRY" condition is encountered (2 & 3)	Same as above	Same as above
	All others (1)	Keystrokes processed normally. These keystrokes are fixed length messages and processing does not look for terminator key	Keystrokes processed normally. These keystrokes are fixed length messages and processing does not look for terminator key	Same as above

(1) MSG RESET and ACK not affected. These keyboard entries are packed as bits in the poll headerword.
(2) An "ILLEGAL ENTRY" condition data value is out-of-limits, invalid data format, ITEM number does not allow data, ITEM number not defined for active SPEC.
(3) ITEM entires in which the keystroke count is exactly 21, 22, or 30 will encounter an "ILLEGAL ENTRY" condition after the last data item (which will be zero).

4.6.1.2.2 MCDS status response.-



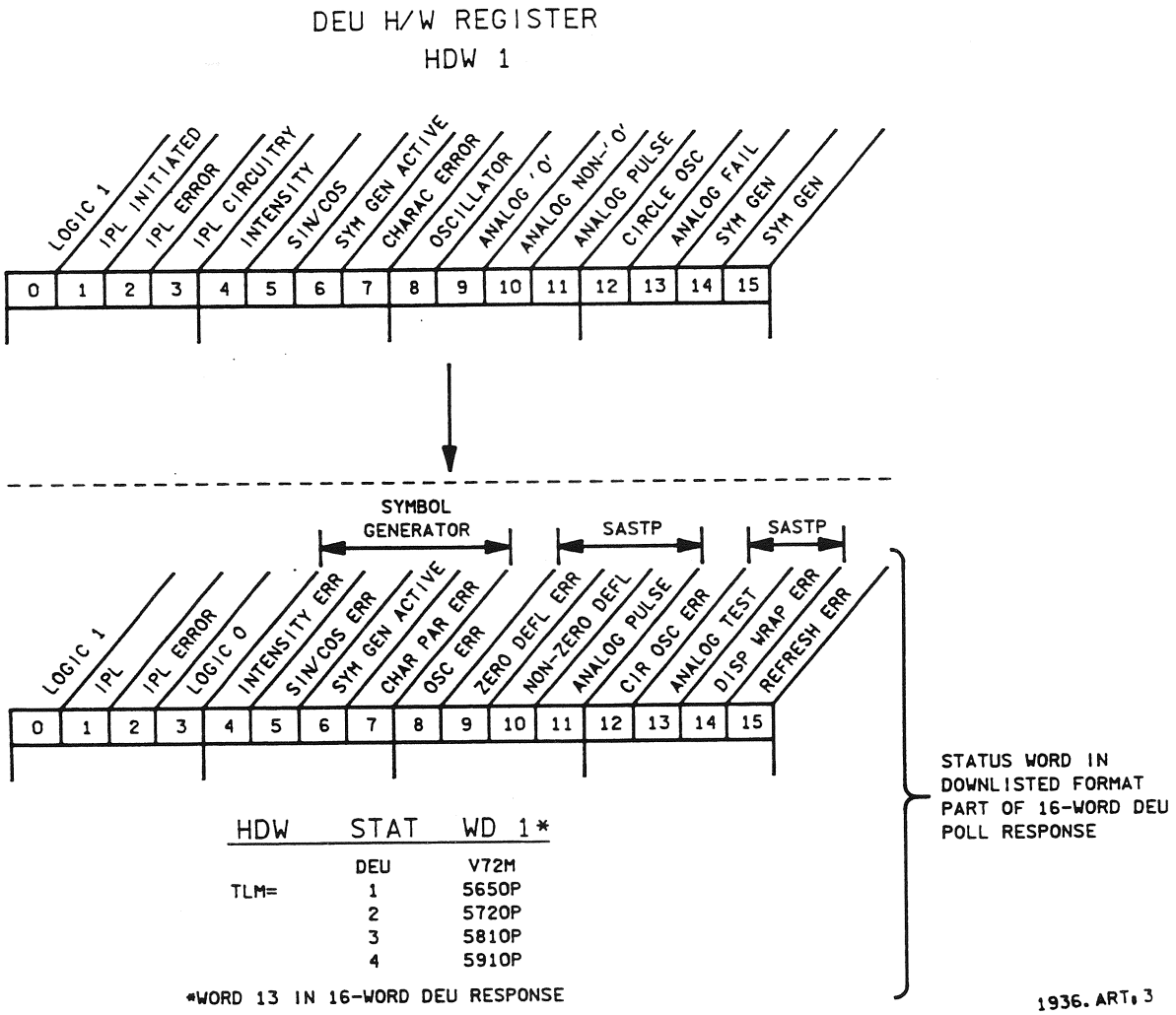
NOTE: Following DEU transmission of this response, bits 13, 19, 21, and 23 of the first data word are reset. Additionally, the count of keystrokes and the hardware and software status words are reset. Bits 9 to 24 correspond to bits 1 to 16 of the downlisted poll header word.

4.6.1.3 DEU Status Words

One small ideocyncracy should be noted in reference to the DEU status words. When the MCDS is powered on and commanded by a GPC with downlist active, the first read of the DEU BITE registers may contain indications of DU-related errors.

These errors will only be visible in the downlist and have no effect on the CRT display or the function of the DEU. Any display irregularities or DU-related errors which occur within 2 minutes of poweron do not indicate an anomaly. If the DU BITE flag trips, the user should verify that it can be reset. No other response is required.

Formation of Hardware (H/W) Status Word 1



HDW STAT WD 1*

	DEU	V72M
TLM=	1	5650P
	2	5720P
	3	5810P
	4	5910P

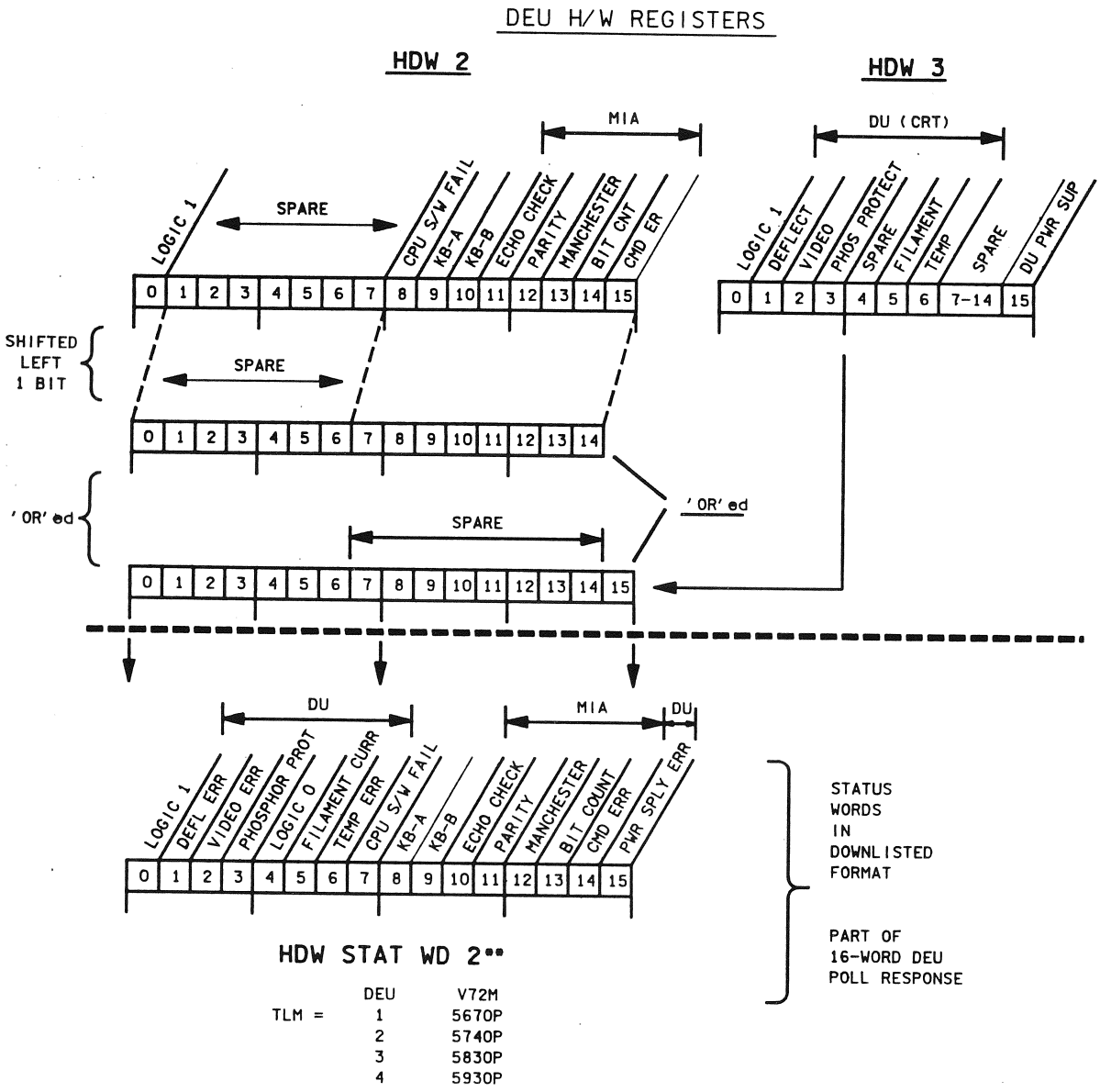
*WORD 13 IN 16-WORD DEU RESPONSE

1936.ART.3

4.6.1.3.1 HARDWARE STATUS WORD 1.- (H/W STAT REG 1)

Bit	Discription
0	Logic 1 - Always set to 1.
1	IPL has been performed - IPL PROM is control of DEU.
2	IPL error - Error is detected during the parity check of the IPL PROM and the multiplexers. The error can occur as a result of an IPL, or as a result of the periodic IPL circuits error check initiated by DEU resident software.
3	IPL circuit check error - (Not currently implemented.) - Error is detected during the parity check of the IPL PROM. The error can occur as a result of an IPL, or as a result of the periodic IPL circuits error check initiated by DEU resident software or during actual IPL.
4	Symbol generator intensity parity error - Parity error is detected on the intensity PROM. The check is performed every time the PROM is activated (i.e., for all symbols). The parity check is not performed when custom intensity is used (custom intensity is not currently being implemented).
5	Symbol generator sine - cosine parity error - Error is detected during the parity check of the sine-cosine ROM of the digital rotation circuit.
6	Symbol generator active - Symbol generator is processing the contents of the refresh buffer. This bit does not indicate the presence of an error condition.
7	Symbol generator character parity error - Error is detected during a parity check of the character generator ROM and data register. The check is performed each time the ROM's read.
8	Oscillator error - Error is detected in the oscillator and the associated countdown logic. The test is performed using a Resistor-Capacitance (R-C) circuit.
9	SASTP: Symbol generator analog zero deflection test error - Zero test of analog deflection is failed.
10	SASTP: Symbol generator analog non-zero deflection test error - Non-zero test of analog deflection is failed.
11	SASTP: Symbol generator analog pulse test error - Pulse has not been detected during an analog pulse test.
12	Circle oscillator error - Circle repetition rate goes to zero.
13	SASTP: Symbol generator analog test error (or of bits 9, 10, 11) One of the failures signified by bits 9, 10, or 11 occurs.
14	SASTP: Symbol generator display wrap error - DEU software detects a no compare on display data that has been wrapped back into memory. The test and the setting of the bit is under software and (digital test) FCW control. The software repetitively issues a Direct Out Instruction to an R-C timing network to inhibit setting this bit.
15	Symbol generator refresh error - R-C timer indicates that the refresh is overdue by 18 ms.

Formation of H/W Status Word 2



HDW STAT WD 2**

	DEU	V72M
TLM =	1	5670P
	2	5740P
	3	5830P
	4	5930P

**WORD 14 IN 16 - WORD DEU RESPONSE

1935. ART, 7

4.6.1.3.2 HARDWARE STATUS WORD 2.- (H/W STAT REG 2, 3 combined)

Bit	Description
0	**Logic one - Always set to 1.
1	**DU deflection error - Improper beam position detected by comparing deflection signal versus yoke return. This error requires arming by the SASTP before it will trip the DU BITE flag.
2	**DU video error - Last stage video amplifier always ON. Circuitry requires performance of SASTP to be functional.
3	**DU phosphor protect error - (not used in OV-102 and subs). Following errors will result in disabling the electron beam (i.e., blank screen) via the G1 control grid: a. Video signal detected with no X or Y beam deflection (small circles can sometimes cause this error). b. Excessive voltage detected in video signal. c. Loss of or steady X or Y deflection current detected. d. Deflection signal detected exceeding 5 percent of the 5-inch by 7-inch viewing range (i.e., off screen beam positioning). e. Power supply error detected - same as bit 15. This error requires arming by the SASTP before it will trip the DU BITE flag.
4	**Spare
5	**DU filament current error - Low filament current detected. This circuitry does not monitor the STBY filament warmup current. Also, this bit does not latch in the DEU hardware BITE register. This error requires arming by the SASTP before it will trip the DU BITE flag.
6	**DU temperature error - One of the temp sensors on the DU deflection amplifiers (two sensors in each X and Y amplifier) or the voltage control page goes above 125° C. This bit does not latch in the DEU hardware BITE register, but will trip the DU mechanical flag (does not require arming by SASTP).
7+	*CPU software fail - Software fails to issue a Direct Out Instruction to the go/no-go timer because software is "hung up" or it skips the instruction. SASTP tests this BITE by momentarily inhibiting the instruction.
8+	*Keyboard Channel A Fail - BITE logic detects a failure in Keyboard Adapter Channel A. If a keyboard message is not transferred to DEU memory within 9 ms the bit is set. The adapter will not send the keyboard message until it receives a strobe from the CPU.
9+	*Keyboard Channel B Fail - BITE logic detects a failure in Keyboard Adapter Channel B. If a keyboard message is not transferred to DEU memory within 9 ms the bit is set. The adapter will not send the keyboard message until it receives a strobe from the CPU.
10	**MIA echo check error - Transmit NRZ data does not compare to the received NRZ data (on the interface between the Singer MIA and the MIA adapter). This check is performed on a bit-by-by basis and only on words transmitted from the DEU to the IOP. The Singer MIA is in the wraparound mode for this error check.
11	*MIA parity error - MIA detects even parity on words detected on the DK bus. This is a Singer MIA-detected error.
12	*MIA Manchester error - Nonvalid Manchester condition occurs on words on the DK bus. This is a Singer MIA-detected error.
13	*MIA bit count error - Word with an incorrect number of bits is detected on the DK bus. This is a Singer MIA-detected error.
14	*MIA command error (or of bits 11, 12, 13) - One of the failures signified by bits 11, 12, or 13 occurs.
15	**DU power supply error - Out-of-tolerance condition detected in the ±80, ±25, -15, or +5 voltages supplied by the low-voltage power supply (voltages are summed then compared to a reference voltage). This error requires arming by the SASTP before it will trip the DU BITE flag.

* Hardware BITE Register 2.
** Hardware BITE Register 3.
\+ Critical BITE error.

4.6.1.3.3 Software status word.-

Bit	Description
0	Display/format data fill error - Presence of a data fill error including MIA errors encoutered during fill, invalid word count (see bit 3), invalid address (see bit 12), and received message imcomplete (see bit 13).
1	Spare
2+	Initialization performed - DCP startup performed. Startup is initiated by an IPL, ITEM A EXEC, or a power transient of less than 15 volts for greater than 400 μs. At initialization, the DCP performs a checksum compare, clears keyboard and SPL, and blank screens.
3	Invalid fill/dump word count - Actual number of fill words received does not agree with specified word count or word count not valid for specified buffer (i.e., <0 or >510 for display buffer). Word count in dump command is invalid (i.e., <0 or >510).
4	MIA wrap word error - For every MIA transmission (DEU to GPC), the headerword transmitted to MIA from DEU memory is wrapped back for comparison. The comparison is made following the MIA transmission, and bit is set for a noncompare.
5	Command overload - A command interrupt received before a last-word-stored interrupt (associated with previous command) received. Typically, a GPC command word received when a data word expected.
6	Spare
7+	CPU memory parity error - Parity error on memory readout. Parity is checked on each word accessed from memory. DEU Incorporates odd parity on words stored in memory. Also set (along with bit 14) when keycode compare fails.
8	Spare
9	Invalid message received - Message OP code in command headerword invalid. Valid codes are those associated with status request (poll), time fills, SPL reset, display fill, format fill (critical formats and SASTP only), memory dump, and buffer fill (buffer fill command is no longer used and its corresponding DCP code is being used as patch space).
10	SASTP: Ripple test error - During SASTP, parity is checked for all memory locations. Then, selected locations are rechecked to verify that parity was correctly restored when each word was written back into memory. Bit is set for a parity error during the recheck.
11	Checksum error - Memory "checksum" error during DCP startup (see bit 2) or critical format load.
12	Invalid fill/dump data address - Starting address is not in allocated buffer or fill goes past end of allocated buffer. Address in dump command not in DEU memory.
13	Received message imcomplete - A GPC message is signaled by a command interrupt followed by a last-word-stored interrupt when the complete message (including data words, if any) is transferred from MIA to DEU memory. The bit is set when the last-word-stored interrupt does not occur within 36 ms following a command interrupt.
14+	CPU self-test error - Computed data noncompare to precomputed values in arithmetic logic unit. Divde, multiply, or branch, subtract, compare, and shift operations are periodically performed. Also set (along with bit 7) when keycode compare fails.
15	Imcomplete DEU transmission - New GPC command received before the last word of a DEU response (DEU to GPC) sent to MIA. The DEU response is the result of a prior GPC command (i.e., poll or memory dump request).

+Critical BITE error.

4.6.2 MCDS Displays

The displays processed by the DEU and CRT receive the dynamic part of foreground data via display updates from the GPC's. The source of the display backgrounds or skeletons is from one of three areas. Certain display backgrounds are resident in DEU memory and are loaded at IPL by PASS GPC's (BFS does not load or use these backgrounds - ref. paragraph 4.6.3 and table 4-IV). Another source of display backgrounds is the GPC. Many PASS display backgrounds and all Backup Flight System (BFS) display backgrounds are sent to the DEU at the initiation of a display. The third source of display backgrounds is the Mass Memory Unit. These displays are referred to as Roll-in SPEC's. See DPS Console Handbook, section 4.8, for a summary of which displays are DEU, GPC, or MMU resident.

4.6.2.1 SPEC/ROLL-IN DISP Restrictions

Presently, system software supports only two SPECs per major function (MF) and/or two ROLL-IN DISPs per MF. There is no restriction on the number of non-ROLL-IN DISPs, but procedurally we never allow all CRTs to be actively driven from one MF; therefore, only two non-ROLL-IN DISPs per MF will be active. Any attempt to support greater than two SPECs/ROLL-IN DISPs per MF will result in an "ILLEGAL ENTRY" message displayed on the CRT from which the request was made.

The ROLL-IN DISP capability is presently implemented only for the SM MF. With this capability, eight SM DISPs and all the CREW TEXT DISPs can be called.

When these restrictions and certain CRT/GPC failure cases are integrated together, the viewing capabilities of the system depend on how software has interpreted the failure. Anytime the GPC/CRT interface fails, the software does not cancel the active SPEC/DISP, but still counts the "lost" SPEC/DISP against its MF limitations. In other words, if the crew had been viewing a SPEC or a ROLL-IN DISP on a specific CRT and that CRT now has an "I/O ERROR" or "BITE" FSP message annunciated against it, the S/W assumes that

the SPEC/ROLL-IN DISP is still active. As a matter of fact, the "lost" SPEC/ROLL-IN DISP can be called up on another CRT; however, the S/W will not release the SPEC/ROLL-IN DISP from the failed CRT.

To work around these restrictions, several recovery techniques are available.

A. At any OPS transition involving a memory configuration change; i.e., G2→G3, all SPECs DISPs, and ROLL-IN DISPs are canceled. As a subset to this, if any SPEC is major mode dependent, then proceeding to another major mode will also cause a cancellation request to be issued; e.g., MM 301 supports SPEC 21 and 22, but MMs 302 and subsequent do not.

B. It is possible for the crew or ground to make S/W think that it heard the RESUME key input from the failed CRT to recover a "lost" SPEC. The crew procedure involves a READ/WRITE technique, whereas the ground can uplink a resume request. Since the crew did not believe the R/W technique was acceptable, the ground will uplink a resume request via DEU EQUIVALENT providing all the following conditions are met:

    1. Ground has AOS and uplink capability to the GPCs.

    2. Affected DK bus must be assigned to the active PASS GPC from which the SPEC/ROLL-IN DISP was called.

    3. The uplink command must exactly duplicate the failed CRTs MF S/W setting that is seen on downlist.

C. To remove the limitation of two ROLL-IN DISPs being active after a GPC/CRT interface failure, the ground using DEU EQUIVALENTS which obey the same commanding criteria as above, will uplink a non-roll-in DISP request to the failed CRT and then uplink a resume request.

During the active entry phase, four SPECs (GPC MEM, RCS, HORIZ SIT, and OVERRIDE) and 0 ROLL-IN DISPs are available to the crew. Potentially, we could trap one of these SPECS and thus be restricted to only one additional SPEC. Normally, the crew uses the HORIZ SIT and OVERRIDE SPECS, but uses the other two in failure cases. Thus, if we trap either the HORIZ SIT or OVERRIDE SPEC, the ground should do nothing. If GPC MEM or the RCS SPEC is trapped, the ground should uplink a resume to the failed CRT.

4.6.2.2 SPEC/DISP Allocations (Revised)

The following table identifies whether a SPEC/DISP is retained or canceled in the GPC software depending upon the user action:

TABLE 4-II.- SPEC/DISP ALLOCATIONS

User action(a)	SPECs	DISPs
RESUME KEY	C	C
SPEC called	C	C
DISP called	R	C
GPC/CRT key reassign - PASS	R	C
- BFS	C	C
BFC CRT switch reassign - PASS	R	C
- BFS	C	C
MF switch change(b)	R	C
OPS transition	C	C
OPS MODE recall - PASS(c)	R	C
- BFS	R	R
MAJOR MODE transition(d)	R	R

(a)Action/results apply to both PASS and BFS unless otherwise noted.

(b)Table reflects effects of SM, GNC, and PL MF in the PASS and SM and GNC MF in the BFS, only. PL MF processing in the BFS results in the following:

- MF from SM to PL while assigned to BFS - previous display retained whether it is SPEC or DISP. Display remains when switch taken back to SM.

- MF from GNC to PL while assigned to BFS - CRT display blanks except for the time fields, SPL, and Msg line. OPS display is presented when switch taken back to GNC, so both SPECs and DISPs are canceled for this case. There are other peculiarities about PASS and BFS processing of the PLMF switch position as follows:

  - In the PASS, when the PLMF position is selected and there is no PL 9 GPC in the common set, the GPC memory display is driven on the CRT (i.e., unsupported MF).

  - In the BFS, if a CRT has a PLMF when the BFS first polls the DEU, the CRT display is blank except for the time fields, SPL, and Msg line.

(c)If GPC target set is altered, it works like OPS transition.DISPs are canceled if there is any change in the NBAT (strings, CRTs, MMUs, launch, etc.) since the previous OPS transition/OPS mode recall. If no change, the DISP is retained.

(d)SPEC/DISP retained only if supported in new major mode.

4.6.2.2.1 General notes.-

A. Active SPEC/DISPs must be canceled on each CRT they are being viewed on.

B. Keyboard ITEM entries apply only to the display currently on a CRT, and therefore, never affect underlying SPECs.

C. For PASS system level SPEC allocation, a GPC can support more than one MF at a time. However, the same SPEC cannot be called up in two different MFs for the same GPC. For example, if the MF switch for a GNC CRT is placed in PL and a systems level SPEC (GPC MEMORY, DPS UTILITY, TIME) is called, then the same SPEC can no longer be called on a GNC CRT. The result would be an "ILLEGAL ENTRY" fault msg. To clear the problem, the MF switch for the appropriate CRT must be returned to PL and the SPEC RESUMED. This scenario does not apply to the BFS. (See User's Guide 3.3 and DRs 25396 and 37515.)

4.6.2.2.2 Sources.- Level A (para. 5.3.1.4), BFS PRD (para. 3.3.1.4), R 19 user's guide (para. 3.3), and SMS checkouts.

4.6.3 DEU IPL

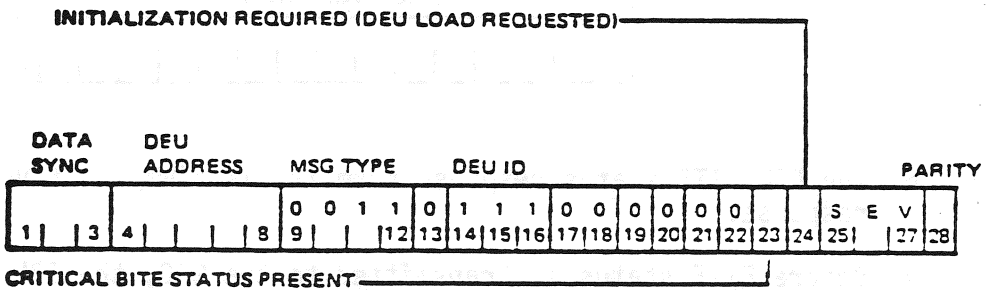
DEU IPL is initiated by engaging the appropriate DEU load switch on panel O6. The DEU load switch is normally in the down position and only requires momentary contact in the up, or load, position to begin the DEU IPL process.

This action results in:

A. The DEU will load its own internal IPL PROM firmware into the DEU memory, resulting in the overlay of any software presently residing in DEU memory. At this time, the DEU IPL PROM software will be in control of the DEU and will be functionally capable of responding to "poll commands" from the GPC. The PROM software will also set the "Initialization Required" bit in the IPL unique "poll response header word." This indicates to the GPC that DEU loading is desired.

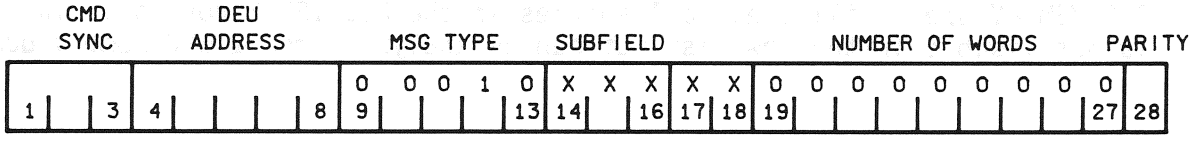
B. A DEU-annunciated message DEU LOAD (lower left hand corner) on an otherwise blank CRT will indicate that the load switch has been activated.

DEU IPL poll status response



NOTE: This one-word DEU response is only used during DEU IPL. Normal DEU to GPC response consists of 16 words, including BITE status words.

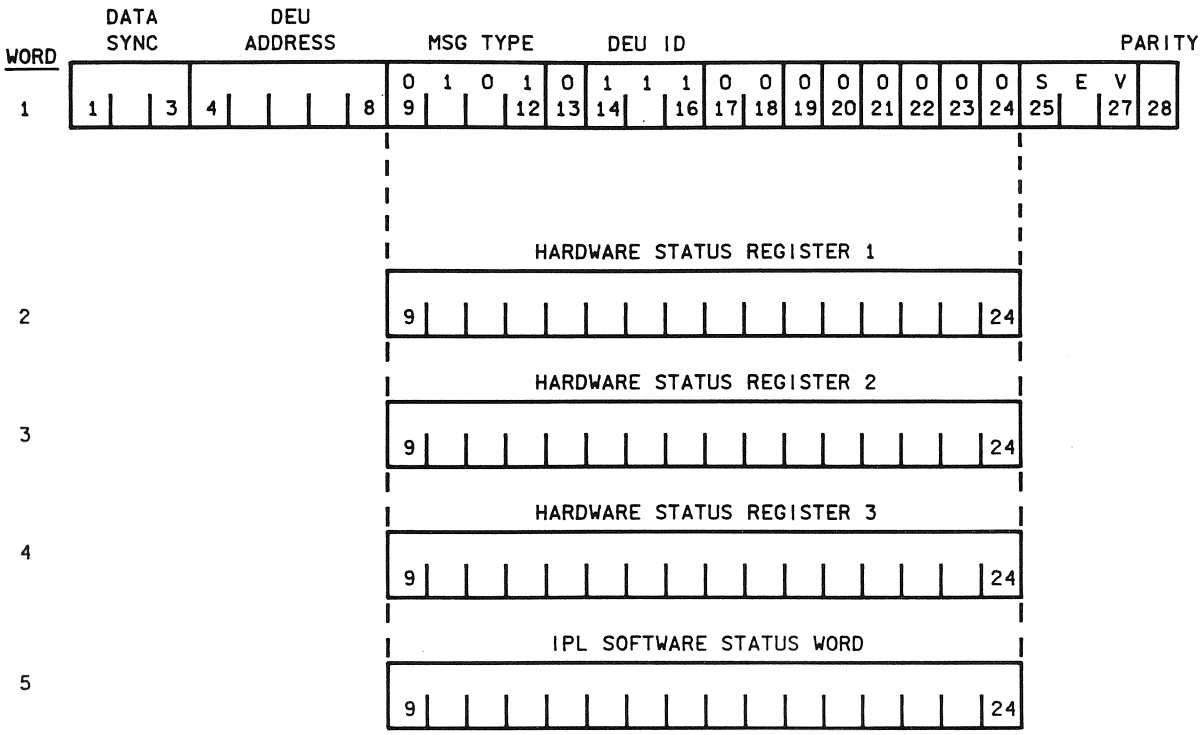
GPC BITE STATUS REQUEST



1950.Art 2

In order to obtain BITE status information during DEU IPL, the GPC issues a BITE status request. Note that this command is only utilized by the GPC during DEU IPL. The DEU responds to this request by sending the contents of the three hardware BITE registers and the IPL software BITE status word. Table 4-III indicates the bit availability for this response. The GPC uses this response to assess the health of the IPL PROM program before continuing with the loading of the DCP.

DEU BITE STATUS RESPONSE FROM IPL



NOTES
- The DEU BITE status response from IPL is always a five-word response.
- Before BITE status is transmitted to the GPC, the IPL PROM (ROM) resets a bit in a mask word. When this occurs, the DEU hardware resets the SEV bits to a 101 pattern.

The GPC maintains a circular log of DEU IPL attempts, indicating the time of the IPL request, the DEU number, and an IPL completion code which describes the error condition, if any, during the IPL attempt. The log contains the first five and the latest five errors. An error log reset (item 48 on the GPC MEMORY spec) will clear all entries in the DEU IPL Error Log. This log is not visible in the downlist but can be analyzed from a GPC memory dump.

TABLE 4-III.- IPL BITE STATUS AVAILABILITY

IPL Software Status

Bit	Meaning when set
0 to 6	Not used
7	CPU memory parity error
8	Not used
9	Invalid message received
10 to 15	Not used

Hardware Status Register 1

Bit	Meaning when set
0	Logical 1
1	IPL has been performed
2	IPL error
3 to 7	Not used
8	Oscillator error
9 to 11	Not used
12	Circle oscillator error
13 to 15	Not used

Hardware Status Register 2

Bit	Meaning when set
0	Logical 1
1 to 7	Not used
8	CPU software fail
9 to 10	Not used
11	MIA echo check error
12	MIA parity error
13	MIA Manchester error
14	MIA BIT count error
15	MIA command error

Hardware Status Register 3

Bit	Meaning when set
0	Logic 1
1	DU deflection error
2	DU video error
3	DU phosphor protect error
4	Not used
5	DU filament current error
6	DU temperature error
7 to 14	Not used
15	DU power supply error

The reaction of the GPC to the "initialization required" bit depends upon which software is currently resident in the GPC. There are three sets of software which have the capability of loading the DEU:

Software	Availability
1. STP/RTP software	• During GPC IPL
2. PASS DEU loader	• Post GPC IPL OPS 0 (MC 0) and PL9
3. BFS DEU loader	• All BFS OPS

The actual DEU software consists of three basic parts and is available from the MMU in three load blocks. Note that the critical formats and the SASTP reside in the same memory locations and are, therefore, never in the DEU at the same time.

MMU LOADBLOCK	SOFTWARE
1	DCP
2	CRITICAL FORMATS
3	SASTP

The current state of software running in the GPC will determine how much of a DEU load the DEU will receive.

Loadblock / GPC software	# 1: DCP	# 2: CRITICAL FORMATS	# 3: SASTP
STP/RTP (GPC IPL)	Yes	Yes	Yes
PASS DEU loader (post GPC IPL OPS 0 (MCO) and PL9)	Yes	Yes	No
BFS DEU loader (all OPS)	Yes	No	No

DCP - The DCP provides the operating software which controls the DEU functions.

CRITICAL FORMATS - The critical formats provide the CRT display skeletons for a number of Primary Avionics Software System (PASS) displays.

Note that because the BFS does not use these format skeletons, the Critical Format loadblock is not included in the BFS DEU load.

TABLE 4-IV.- CRITICAL FORMATS - DEU RESIDENT DISPLAYS PER OI-21 PASS USERS GUIDE

Display page no.	Display title	Applicable memory config.	Format index (bits 9-16 poll response wd 2	Format index table entry
099 [a]	FAULT	ALL	00	1
050	HORIZ SIT	1, 3	01	2
305, 602 603	VERT SIT 1(2)	1, 3	02	3
104 to 106 202 301 to 303	XXXXX MNVR YYYYY	1, 2, 3	03	4
101 to 103, 601	XXXXX TRAJ	1	04	5
018	GNC SYS SUMM 1	1, 2, 3, 8	05	6
304	ENTRY TRAJ 1(2,3,4,5)	3	06	7
000 [a]	GPC MEMORY	All	07	8
022	S TRK/COAS CNTL	2, 3	0B	12
021	IMU ALIGN	2, 3	0C	13
051	OVERRIDE	1, 3	0D	14
023	RCS	2, 3, 8	0F	16
NONE [b]	Spare		10 - 1D FF	17 - 30

[a] System level displays
[b] For all displays not stored in DEU

SASTP - The SASTP is a comprehensive DEU and DU self-test routine. It should be noted that:

A. Available only during GPC IPL (STP/RTP S/W)

- Item 20, IPL menu display

B. SASTP overlays the critical format software when it is loaded into DEU memory.

C. SASTP is temporary, since critical formats are reloaded into DEU memory whenever SASTP is terminated.

SASTP is discussed in more detail in the next section.

IPL COMPLETE - Upon successful completion of the DCP Load, the DCP performs a checksum on the DEU memory, and compares the result to a stored value. If they are equal, the DEU-annunciated message IPL COMPLETE will be displayed. It should be noted that the process of a DEU load may be so fast that the OPS 0 or IPL menu displays will appear on the CRT so quickly that the DEU messages DEU LOAD and IPL COMPLETE may not be seen. However, the appearance of the OPS 0 or IPL menu display will definitely be an indication that the DEU load was successful.

IPL INCOMPLETE - If DEU loading software is not available when the "initialization required" bit is set, the GPC will stop further processing of the particular poll response and will not annunciate an error message. The only message present will be the DEU-annunciated message of DEU LOAD which indicates that the load switch has been activated. Unfortunately, when a PASS GPC is incapable of loading a DEU, it continues nominal transactions to the DEU (time fills, polls, display updates). The DEU IPL software mistakes time fills for its anticipated load and fills its memory with the time fills until it destroys itself. If this happens, the DEU LOAD message is lost and the screen is totally blank. To regain the use of the DEU, a GPC-to-DEU interface must be established in which GPC DEU loading software is available (i.e., re-IPL a GPC, transition to PL 9, or transfer the DEU to the BFS.

Inadvertent DEU IPL - When a DEU IPL is attempted from a PASS GPC other than PL 9 or post IPL OPS 0, the associated CRT will probably blank but no fault message will be annunciated. If an illegal DEU IPL is suspected, the console operator should look at the POLL HDR WD (poll header word) on the appropriate DEU STATUS history tab (1359, 1360, 1361). A value of 370X (X can be 0-3) verifies that a DEU IPL has been attempted (see attached example). For a definition of the specific bits in the poll header word, see SCP 4.8.

4.6.4 SASTP

The SASTP is a comprehensive MCDS self-test routine available only during GPC IPL (it can only be activated via item 20 on the IPL MENU display). Contact with the GPC during SASTP is nominal GPC-to-DEU polling. SASTP will test the following DEU functions:

A. Display generation

B. Major function switch operation

C. Keyboard A/B inputs

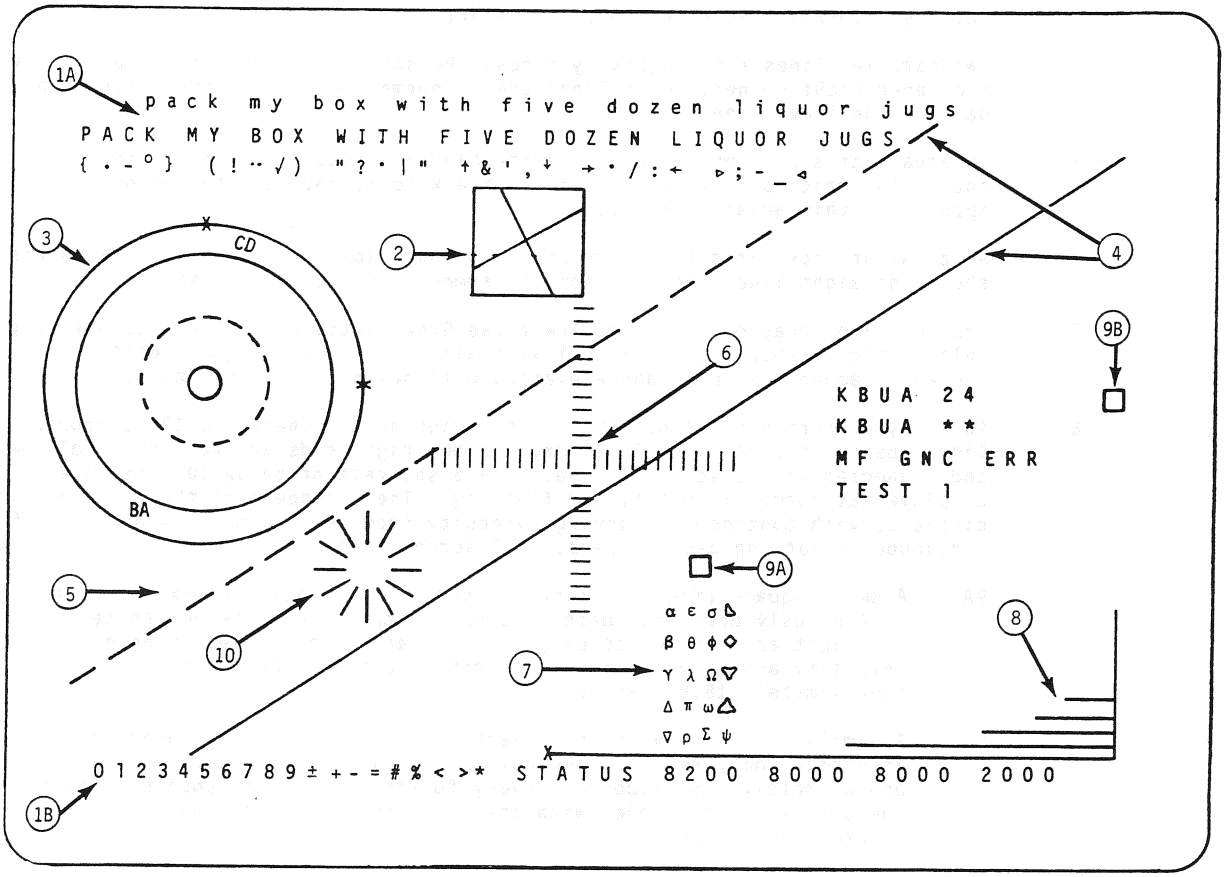


Figure 4-29.- SASTP display.

SASTP Display Description:

1 DEU symbol elements.

1A The first line displays lowercase Roman letters, second line displays uppercase Roman letters, and third line displays characters, all in normal size and intensity.

1B Numerals are displayed in normal size and intensity. Mathematical symbols are displayed at bright intensity and normal size.

2 A square with sides 0.7383 inch in length, oriented horizontally and vertically and drawn with normal intensity. Within the square is a dynamic display of rotating vectors, counterclockwise.

3 Display of four concentric circles in normal intensity. Three are drawn in solid lines and one is drawn with dashed lines. A capital letter X, at normal intensity, is written so its center coincides with the largest circle at the top (0°). Another such capital X is on the largest circles at the right side (90°). Within the arrangement of circles is a dynamic display of rotating and revolving letters, clockwise in direction, 180° apart.

4 Two parallel lines run diagonally across the screen from the lower left corner to the upper right corner. Both lines are of normal intensity, the upper line is dashed, the lower line is solid.

5 An area that should be blank unless the blanking feature is malfunctioning. If the letter writing is working, but the blanking is not, the following legend appears in this area: BLK FAIL.

6 An array of tick marks for checking focus and resolution. The tick marks are short, straight line segments from the symbol generator character matrix.

7 Four-by-five array of upper and lowercase Greek letters, del (upside down capital delta), tacan symbol (an outline Y with all three arms of equal length and angle), diamond, and plan and elevation outlines of the Space Shuttle.

8 An array of horizontal lines of varying brightness. There are 10 horizontal lines spaced vertically equidistant, and the right ends are vertically aligned and connected with a vertical line. The 5 shortest of these 10 lines are displayed at normal intensity and flashing. The 5 longest of these 10 lines are displayed with continously varying intensity from zero to maixmum and continuously varying back to zero, at 3-second intervals.

9 9A A small square (from the symbol generator character matrix) which travels continuously back and forth in a horizontal line (between the center and the right edge of the screen). The square is displayed in normal intensity and travels back and forth, a complete cycle requiring approximately 18.62 seconds.

9B A similar vertically moving square. This square is displayed in normal intensity and travels up and down, a complete cycle requiring approximately 4.63 seconds. Every fourth cycle, this square coincides at the bottom of its travel with the horizontally moving square at it rightmost position.

10 A spinning "bug" consisting of 16 lines in a sunburst array, rotating about a center which moves up and to the right. The pattern does not appear to be a rolling wheel, but rather an emblem at the center of a rolling wheel, with about four times the diameter. Immediately following the upward transit, the bug reverses direction and appears to roll and slide down the hill at the same rate it moved up. Then it starts up again. The complete cycle takes approximately 26.48 seconds.

SSR: SASTP

SASTP is rarely used. The following procedure was developed (in MALF- SSR format) to enhance SASTP understanding.

NOTE: This procedure is not available in the Flight Data File.

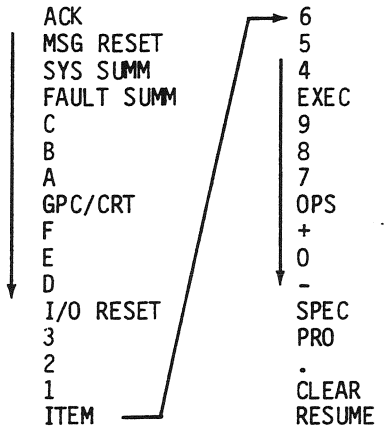
1. |IPL MENU|
   Activate SASTP:
   ITEM 20 EXEC

2. On lower right of SASTP display:
   ✓BITE status registers appear after 12 seconds
   ✓Status = 8200 8000 8000 0000

3. Observe dynamic movements on display

4. ✓Major Function Switch operation:
   If MF sw = GNC, set MF sw - SM, then set MF sw - PL
   If MF sw = SM, set MF sw - GNC, then set MF sw - PL
   If MF sw = PL, set MF sw - GNC, then set MF sw - SM

5. ✓KYBD Inputs:
   Depress KEYS in the following order:



6. ✓Forced BITE Error Test
   Item 1 EXEC

   On lower right of SASTP display:
   ✓STATUS = 8276 8080 A000 0000
   ✓on CRT:
   Mechanical BITE Flags tripped and reset Flags

   |SASTP|
   ✓No '*' by TEST 1, then
   ITEM 2 EXEC
   On lower right: ✓STATUS = 8200 8000 8000 0000

7. Return MF switch to ORIGINAL position, then enter RESUME

8. ✓|IPL MENU|

Unfortunately, during normal execution of the SASTP, the ground is not processing data from the DEU (it is assigned to a GPC undergoing IPL). However, if the need arises to look at a CRT undergoing SASTP from the ground, the following procedure can be employed:

A. Perform MENU IPL of a PASS GPC using affected CRT

B. IPL MENU

   Activate SASTP: ITEM 20 EXEC

C. Take BFC CRT select switch to OFF - this results in the GPC IPL process to default to a PASS default IPL.

D. Assign CRT to PASS and monitor BITE words

E. Assign CRT to IPL'ed GPC and reload the CRT

4.6.5 OTP

The Operational Test Program (OTP) is a collection of test capabilities that can be used to test/troubleshoot DEU hardware and software. OTP does not require a GPC to DEU interface and is available whenever the DCP is in control. The test functions can be utilized individually or in any multiple combination. The following actions are available with the OTP:

Test function	Keystroke sequence
1. Initizalize DCP	ITEM A EXEC
2. BITE STATUS word display	ITEM B EXEC
3. Clear BITE register	ITEM C EXEC
4. Dump DEU memory (where XXXX is a hexadecimal number)	ITEM D + XXXX EXEC
5. Exit OTP	ITEM E EXEC
6. Format review (where XX is a decimal number from 1 to 30)	ITEM F + XX EXEC

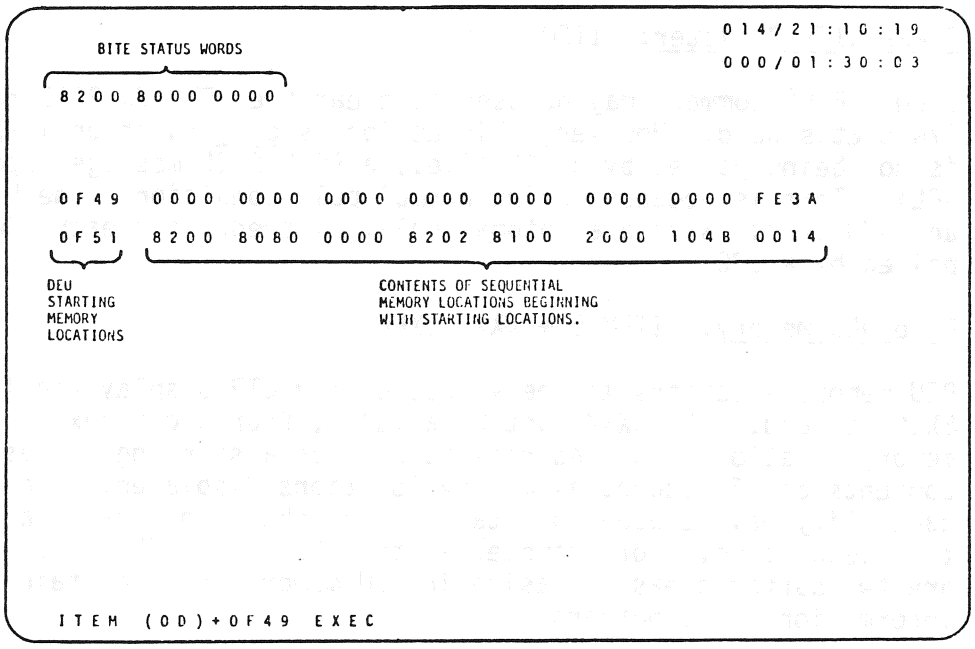


Figure 4-30.- DTP display.

NOTE: The SPL and "Time Fields" are functional on this display. The big "X" which indicates a loss of display update is not supported during OTP.

4.6.5.1 OTP Functions

A. Initialize DCP: ITEM A EXEC

To initialize the DEU, ITEM A EXEC is entered via the keyboard. Initializing the DEU constitutes a simulated power transient. The DEU responds by:

1. Blanking the current CRT display
2. Clearing the keyboard entries and SPL
3. Performing a checksum of the DCP
4. Setting certain bits in the H/W and S/W status words to reflect - 8202 8100 2000

NOTE: If the OTP display or any of the format review displays are on the CRT, they will automatically reappear following an "ITEM A EXEC."

B. BITE STATUS Word Display: ITEM B EXEC

ITEM B EXEC command results in three BITE STATUS words being displayed. From left to right, they are: H/W stat word 1, H/W stat word 2, and S/W status word. The status words are dynamic and reflect any changes in the DEU H/W status registers or the DEU S/W status word.

C. Clear BITE Register: ITEM C EXEC

ITEM C EXEC command may be used to clear the DEU H/W BITE registers and S/W status word. However, this action is only functional when the DEU is not being polled by a GPC (i.e., a POLL FAIL message appears on the SPL). This is because during normal poll conditions, the H/W registers and S/W status word are automatically cleared/reset each time the DEU is polled by a GPC.

D. Dump DEU Memory: ITEM D + XXXX EXEC

DEU memory locations may be viewed on the OTP display via ITEM D + XXXX EXEC command. The XXXX must be a valid, four-digit hexadecimal DEU memory location. This address is used as a starting address with the contents of 16 sequential memory locations displayed. The DEU dump capability may be used to obtain an insight into more data for problem troubleshooting. For example, in the event of a CRT BITE error, there are two buffer areas set aside in DEU memory which contain valuable information. The buffers are:

- Error key code buffer Loc OF49 → OF 50
- Error register buffer Loc OF51 → OF 56

Due to their proximity to one another in memory, both buffers can be observed via ITEM D + OF49 EXEC; both buffers are discussed in detail in section 4.6.5.2, "DEU memory dump - applications."

The poll response keyboard buffer can also be reviewed at locations OF95 through OFA4 for words 1 to 16 respectively. All addresses are for DCP 8.07.

E. Exit OTP: ITEM E EXEC

Three of the test sequences, BITE STATUS word display, dump DEU memory, and format review, generate their own CRT displays. In order to view the OPS, SPEC, or DISP display that was being viewed prior to initiating OTP, an ITEM E EXEC must be entered via a keyboard. This action will terminate OTP, and the previous OPS, SPEC, or DISP display will reappear on the CRT.

F. Format Review: ITEM F + XX EXEC

ITEM F + XX EXEC, where XX is a valid, two-digit decimal number, will result in the appropriate display background or skeleton's being displayed. There are 16 different displays, or Critical Formats presently residing in the DEU Critical Format buffer. The display skeletons and their respective two-digit format indexes are shown in table 1-II.

If consecutive background formats are called up without first exiting the OTP (ITEM E EXEC), the DEU critical format loads will be lost, and the DEU will have to be re-IPL'd.

4.6.5.2 DEU Memory Dump - Applications

A. Error Keycode Buffer Locations OF49 → OF50

When a key is depressed, the DEU KYBD adapter produces a 16-bit keycode, which is sent to the DCP. When the keycode is received from the KYBD adapter, the DCP (S/W) checks for one of four critical BITE's: (1) KYBD adapter A, (2) KYBD adapter B, (3) CPU self-test, and (4) CPU memory parity error. If one is detected, the keycode will not be processed, but will be saved in the "error keycode buffer," and the flashing "ERR" will be driven on the CRT.

The buffer is an eight-deep pushup stack, such that the latest errored keycode is stored in that last word of the buffer (last-in, last-out).

ERROR KEY CODE BUFFER

ADDRESS:		
OF49	1	LEAST RECENT ERROR
4A	2	
4B	3	
4C	4	
4D	5	
4E	7	
4F	7	
OF50	8	MOST RECENT ERROR

If no critical BITE's are detected, the DCP compares the keycode to a list of legal keycodes found in the keyswitch code table (DCP locations OFB4 → OFD4). If no match is found, the DCP stores the keycode in the "error key code buffer"; the DCP sets both the CPU self-test and the CPU memory parity error critical BITE bits (bits 7 and 14 of S/W status word). If, on the other hand, a keycode match is found, the matching "DEU internal code" or keystroke code is used in transmission to the DEU (words 3 to 12, MCDS status response).

B. Error Register Buffer Locations OF51 to OF56

This buffer is used to store the BITE status words whenever a critical BITE is detected. The critical BITE's are:

1. KYBD adapter A error
2. KYBD adapter B error
3. CPU self-test error
4. CPU memory parity error
5. Initialization performed (or power transient)

The buffer contains the BITE status words for the last two times a critical BITE has been detected. It is a pushup stack with the latest set of errored status words stored in the last three buffer locations (OF54 → OF56).

ADDRESS:			
OF51	1	H/W STAT 1	
52	2	H/W STAT 2	LEAST RECENT SET
53	3	S/W STAT	
OF54	1	H/W STAT 1	
55	2	H/W STAT 2	MOST RECENT SET
56	3	S/W STAT	

The software uses the critical BITE present bit in the poll response headerword (bit 23) to determine when a critical BITE has been observed. The BITE words are stored in the buffer when the BITE registers are cleared following a DEU poll response. Therefore, if there is no polling (i.e., POLL FAIL message) when a critical BITE is detected, the appropriate BITE words will not appear in the buffer until polling is resumed. The current BITE words, however, can always be viewed on the OTP display. A typical set of BITE words in the buffer is 8202 8100 2000 because this is the pattern observed following a DEU LOAD or power transient and has the initialization performed bit set.

4.6.6 MCDS Messages

There are 11 fault messages related to the status of the MCDS. There are two sources for these messages, the DEU itself or the GPC.

A. DEU Annunciated Messages

1. DEU LOAD
2. IPL COMPLETE
3. IPL INCOMPLETE
4. POLL FAIL
5. BIG "X"
6. ERR

B. GPC Annunciated Messages

1. CRT BITE 1(2,3,4)
2. I/O ERROR CRT 1(2,3,4)
3. I/O ERROR MMU
4. > 3 DEU
5. ILLEGAL ENTRY

4.6.6.1 DEU Annunciated Messages

The DEU software is responsible for annunciating six messages:

- DEU LOAD
- IPL COMPLETE
- IPL INCOMPLETE
- POLL FAIL
- Big "X"
- ERR

The first three messages are used to indicate the status of a DEU IPL. They are discussed in detail in the DEU IPL section of this document.

POLL FAIL

The POLL FAIL message is an indication that the DEU is no longer being polled by or receiving CRT mission and event time fills from the GPC. Note that these two commands are chained so that a poll should always follow a time-fill command. If no time-fill command is received by the DEU, or if the poll fails to follow the time fill within 5 ms for a duration of 3 seconds, and if the display is not frozen or IPL is not being performed, the DEU will reset the SPL and display POLL FAIL on the right-hand side of the SPL. Note that because a DEU will process a poll independently of the time fill, a POLL FAIL message may be an indication of only the loss of time fill and not actually the loss of the poll. The DEU will continue to compute mission and event time internally, and will continue to update the time fields on the CRT. While the POLL FAIL message is on the CRT, the DEU will accept keyboard inputs, but will not flag the inputs for transmission to the GPC. The POLL FAIL message will remain on the CRT until the first valid chained time-fill and poll commands are received, at which time the POLL FAIL message will be removed from the display and the DEU will reset the SPL.

Big "X"

The big "X" across the CRT display indicates that the DEU has not received the display data fill command from the GPC. If the display data fill command is not received by the DEU for 3 seconds, and if there is no display freeze (caused by single depression of SPEC key), the DEU will draw an "X" diagonally from corner to corner across the CRT. The mission and event times and the SPL remain functional during loss of display update. The "X" is removed from the display upon receipt of the next display data fill command at which time the DEU performs a display freeze/unfreeze sequence. Also note that the big "X" is cancelled while an OTP display (or format preview) is active.

ERR

The flashing ERR on the SPL is an indication that an illegal keyboard entry has been made. An illegal keyboard entry can be:

- Entry which is illegal by "DEU Syntax Check"
- Entry that exceeds the (ref paragraph 4.6.8) SPL capacity
- Entry while a "critical bite" bit is set

Note that the GPC will clear the SPL upon detection of a critical BITE. Therefore, the flashing ERR during detection of a critical BITE may never be seen. The ERR will be removed upon depression of the CLEAR key or by reinitiating a valid keyboard sequence.

4.6.6.2 GPC Annunciated Error Messages

The GPC software is responsible for annunciating five messages related to MCDS error conditions. The messages are:

- CRT BITE 1(2,3,4)
- I/O ERROR CRT 1(2,3,4)
- I/O ERROR MMU
- >3 DEU
- ILLEGAL ENTRY

CRT BITE 1(2,3,4)

This message is a class 3 fault message indicating that a critical BITE has been set by the DEU. The critical BITE's are:

- CPU software fail
- Keyboard adapter A fail
- Keyboard adapter B fail
- Initialization performed
- CPU memory parity error
- CPU self-test

The GPC checks for critical BITE via bit 23 of the MCDS status response headerword. If this bit is set, the GPC typically annunciates the fault message. There are, however, some peculiarities depending on what GPC S/W is attached to the DEU.

PASS S/W nuances. Whenever the "Critical Bite Present" bit is set (bit 23, DEU response headerword), the PASS GPC will check further to determine if the detected BITE is "initialization performed" (bit 2, S/W status word). The latter indicates that a power transient has occurred. If this bit is set, a PASS GPC will not annunciate the BITE message but will re-initialize the active CRT display. If bit 2 is not set, the message is annunciated. In addition to the error message, a PASS GPC will stop polling the errored DEU but will continue to update the display. Therefore, the DEU will annunciate and display its "POLL FAIL" message. Polling may be resumed by:

- Reassigning the DEU via GPC/CRT key
- OPS transition/OPS mode recall (with a target set change)
- Use of BFC/CRT switch; assign, then release DEU/CRT from BFS

BFS S/W actions. The BFS S/W will annunciate CRT BITE 1(2,3,4) solely on the basis of seeing bit 23 set. Unlike the PASS, the BFS will continue to poll the errored DEU. Therefore, the DEU will not annunciate a POLL FAIL message. The BFS S/W will also continue to update its displays.

I/O ERROR CRT 1(2,3,4)

This is a class 3 fault message annunciated as a result of one of the following (reference DPS Console Handbook table 4.2-XII for a summary of error signatures):

- Bus Control Element (BCE) type errors detected at the Flight Control Operating System (FCOS) level (i.e., DK bus failure, DEU RCVR/TRXMTR failure, or GPC RCVR/TRXMTR failure)

- Checksum error detected at the user-interface level

Regardless of what level of systems software detects the error condition, the user-interface level is responsible for annunciating the message. Note that polling and display updates to the DEU continue when this message is annunciated. I/O error detection details are as follows:

A. FCOS-Detected I/O

The PASS GPC FCOS and BFS GPC BOS S/W will detect both transmitter or receiver errors. Note that when a reception error is detected, the PASS S/W checks further to see if a DEU IPL has been requested; if it has, instead of annunciating the I/O ERROR CRT 1(2,3,4), the S/W will attempt to schedule the DEU IPL. If the DEU IPL is not legal in the present memory configuration, software processing for this DEU is complete for this cycle. The I/O ERROR CRT 1(2,3,4) message is, therefore, never annunciated for this case. The GPC will stop annunciating the I/O ERROR CRT 1(2,3,4) message after the first error detected for sequential errors, and will resume error annunciating only after a DEU message is processed with no detected errors. When an error occurs resulting in the fault message, software processing for this DEU is complete for this cycle.

If a CRT is not deassigned from the GPC with which it is communicating before being powered off, several types of scenarios can be encountered in which initial timeout errors will be annunciated. First, when the CRT is powered off, two I/O errors will be logged by GPCs in the major function to which the CRT is assigned. Similarly, an OPS transition will produce the same number of errors. Finally, an I/O reset will produce multiple errors depending on which buses are assigned to the major function in question. All errors will be initial timeout errors with the following signature:

ID - 5, 6, 7, or 8
EL - 0
BCE - 6, 7, 8, or 9
STAT - 00000040

The I/O errors annunciated in the I/O reset scenario will be discussed in detail below. CRTs are polled continuously (i.e., CRTs are "non-bypassable" elements) by the GPC to which it is assigned even when the CRT is powered off. The first I/O error which occurs when a CRT is

powered off is logged and annunciated. The second I/O error which occurs is only logged. Therefore, after these first two errors, the GPC continues to poll the CRT but does not log or annunciate the failed communication transactions. New CRT I/O error messages will only be annunciated after the first successful poll occurs to the CRT. This is due to the fact that the CRTs I/O error message flag is cleared out only after a successful poll response is received by the GPC. CRT I/O errors can be logged but not annunciated if the ID transaction counters associated with the CRTs are cleared.

An I/O reset by nature of implementation clears out all ID transaction counters upon the attempted upmode of each string/bus pair assigned to the major function in question. In this case, a new fault message is not annunciated because a successful poll has not occurred, but I/O errors are logged due to the cleared transaction counter. For GNC, strings/bus pairs are upmoded starting with flight-critical strings (FC1/5 through FC4/8), then the payload bus pair (PL1 and PL2), and finally the launch data bus pair (LDB1 and LDB2). An I/O reset will only upmode one string/bus pair per major cycle. Hence, a maximum total of six ID transaction counter clears will be processed for a GNC I/O reset. Since the CRTs are being polled at a 2 Hz rate and the ID transaction counters are being cleared at a 1 Hz rate, two errors will be logged against a CRT which has been powered off and has not been deassigned for every string/bus pair upmode attempt. Therefore, the number of strings and bus pairs assigned to the major function which is being reset times two will be the number of I/O errors to expect for a CRT which is powered off and not deassigned when an I/O reset is performed.

B. User-Interface Detected I/O Errors

Checksum processing is performed by the PASS GPC whenever the "Keyboard Message Present" bit or the "Critical BITE Present" bit is set in DEU poll response headerword (bits 21 and 23 respectively) and by the BFS on every POLL response. The user-interface systems software will annunciate an I/O ERROR CRT 1(2,3,4) error message if the poll response fails checksum processing. The GPC will stop annunciating this message after the initial checksum miscompare for sequential miscompares, and will resume error annunciating only after a message is received which passes checksum processing. If the error message is annunciated, the keyboard message and BITE status words in the poll response are ignored. However, major function processing and MSG RESET/ACK processing are executed. Note that no I/O errors are logged for checksum failure, and that the BFS is more susceptible to this annunciation because it performs checksum processing on every poll response.

This page intentionally blank

I/O ERROR MMU

The I/O ERROR MMU message is a class 3 fault message which typically indicates that an I/O error has been detected during an MMU transaction. However, the message may also indicate that an error was detected during initial program loading of the DEU from a PASS GPC. Note that the BFS GPC does not issue an error message for a DEU IPL error. The PASS GPC-annunciated error may occur during any phase of the DEU loading including DEU IPL PROM fill (i.e., detected BITE errors following IPL PROM takeover), MMU read, DEU memory fill, and DCP startup. Note that when a DEU IPL error is detected, the DEU being loaded will not be capable of displaying the error message. The message will appear on common set CRT's only. The errored DEU will display the DEU-annunciated message, "IPL INCOMPLETE," if the error were also detected by the DEU, or it will continue to display "DEU LOAD" if the error were not detected by the DEU.

>3 DEU

The >3 DEU message is a class 3 fault message indicating that the user has attempted to assign a fourth DEU into the common set via a GPC/CRT keyboard entry. The entry will be ignored and the error message generated. The reason for this three-DEU limit in the common set is that PASS S/W cannot handle the I/O for four DEU's (causes cycle wraps).

ILLEGAL ENTRY

The ILLEGAL ENTRY message is a class 5 fault message indicating that an invalid or illegal response has been received from the DEU. This message does not indicate a H/W or S/W failure, but is instead an indication of a user procedural error. The message is also annunciated by applications S/W as a result of inappropriate item entries to applications SPECSs. Note that this error message is not associated with keystroke syntax errors. The syntax errors are identified and annunciated by the DEU. The ILLEGAL ENTRY fault message appears on the affected CRT only and overlays any previous fault messages on the fault message line. See Console Handbook, section 4.8 for a summary of the errors that result in an ILLEGAL ENTRY.

Summary

Fault Message	I/O error log	Problem
I/O ERROR CRT	Initial T-0 followed by Initial T-0	CRT interface problem (i.e., DEU failed/powered off, simplex GPC transmitter/receiver failed, etc.)
I/O ERROR CRT	None	CRT checksum error
I/O ERROR CRT	Initial T-0 followed by Interword T-0	Probably an unintentional DEU load. Check the MF on the appropriate display and see if it is 0 (use DPS 1 if CRT on GNC GPC, DPS 3 if CRT on SM GPC, DPS 4 if CRT on OPS 0/PL 9 GPC). Next check the poll header word on the appropriate history tab. If there is a 370 X, this confirms the DEU load request
None	Interword T-0 followed by Initial T-0	Same as above

4.6.7 Console Tools for Determining CRT/DEU Assignments

The DPS console event lights and high speed displays provide sufficient information to determine CRT/DEU real-time assignments. The intent of this information is to identify the console cues relating to CRT/DEU assignments, and how to interpret them. It is important for the console operator to remain aware of CRT/DEU distribution as it impacts the three CRT/DEU limit in the common set, and because it would be inappropriate for all PASS assigned CRT/DEUs to be assigned to a single PASS GPC. This is because the loss of the single GPC would isolate all PASS CRT/DEUs and undesirable procedures would be required to reestablish a PASS CRT/DEU interface. This note will first describe the pertinent event lights and high speed display parameters driven by PASS GNC and BFS Downlist, then provide examples of specific CRT/DEU assignments with their related console cues.

4.6.7.1 DK Transmitter Enables

The status of the PASS DK Transmitter enables is identified by both the event lights DDDs (nos. 448 and 449) and the high speed displays (XMIT ENA WD1 on DPS 2 - GNC/BFS). The status of the BFS DK transmitter enables is identified only on the high speed displays (BFS XMIT ENA WD 1 on DPS 2 -

GNC/BFS). The specific bits relating to DK transmitter status in the XMIT ENA WD 1 for PASS and BFS are defined as follows:

BITS 1-6	BIT 7	BIT 8	BIT 9	BIT 10	BITS 11-16
	DK 1 ENA IOP PORT 6	DK 2 ENA IOP PORT 7	DK 3 ENA IOP PORT 8	DK 4 ENA IOP PORT 9	

Note that if a CRT/DEU is assigned to the BFS via a GPC/CRT keyboard input (as opposed to a BFC CRT switch assignment), the BFS XMIT ENA WD 1 is the only source which will identify this assignment other than actually seeing the keyboard input on the high speed display scratch pad lines. Also note that the scratch pad lines, DISP, MF, H/W1, H/W2, and S/W Fields on the high speed displays are PASS driven only.

4.6.7.2 BFC CRT Switch Positions

The BFC CRT DISP and SEL switch positions (Panel C3) result in CRT/DEU assignments to the BFS GPC. The switch positions are identified both by BFS driven event lights DDD (no. 439) and high speed displays (PASS and BOS DISC B on DPS 1 - GNC/BFS). The specific bits relating to the BFC CRT switch positions in DISC B for PASS and BOS are defined as follows:

BITS 1-6	BIT 7	BIT 8	BITS 9-16
	BFC DK SEL A	BFC DK SEL B	

Interpretation of the event lights and DISC B bits as they relate to the BFC CRT DISP and SEL switch positions is provided below:

EVENT LIGHT (BIT)		BFS CRT SWITCH	
BFC DK SEL A	BFC DK SEL B	DISP	SEL
OFF (0)	OFF (0)	OFF	—
OFF (0)	ON (1)	ON	1+2
ON (1)	OFF (0)	ON	2+3
ON (1)	ON (1)	ON	3+1

Note that although a CRT/DEU is assigned to the BFS via the BFC CRT DISP and SEL switches, the original PASS commander of the CRT/DEU does not turn off its transmitter. Instead, the PASS GPC sets a bus mask for the CRT/DEU. Therefore, it will appear that two different GPCs are transmitting to one CRT/DEU (i.e. dual commanders) when in fact only the BFS GPC's commands are being heard by the CRT/DEU.

4.6.7.3 DEU On/Standby

CRT/DEU on/standby status is provided by OI driven event lights (ID 438). A CRT/DEU will communicate with a GPC only if it is ON and a GPC is currently transmitting to it. The standby switch position allows power only to the CRT Filament for warmup. Note that although a CRT/DEU may be OFF (or in STBY), a GPC could still be transmitting to it. The GPC would issue a single I/O ERROR CRT 1 (2,3,4) fault message and continue transmitting to the CRT/DEU. Therefore, this CRT/DEU would still be counted in the three CRT/DEU limit in the common set. Conversely, a CRT/DEU could be ON with no GPC attempting to communicate with it. This CRT/DEU would not be counted in the three CRT/DEU limit.

Examples

A. DK Transmitter Enable Lts - GPC 1 ENA DK 1 - ON
GPC 2 ENA DK 2 - ON
GPC 4 ENA DK 3 - ON
all other lts - OFF

BFS XMTR ENA WD 1 -

BITS 1-6	BIT 7	BIT 8	BIT 9	BIT 10	BIT 11-16
	0	0	1	1	

BFC CRT SWITCH Lts - BFS DK SEL A - ON
BFS DK SEL B - ON

DEU ON/STBY Lts - DEU 1 ON - ON
DEU 2 ON - ON
DEU 3 ON - ON
all other lts - OFF

Analysis: CRT/DEU 1 - assigned to GPC 1
CRT/DEU 2 - assigned to GPC 2
CRT/DEU 3 - assigned to BFS via BFC CRT switches
CRT/DEU 4 - assigned to BFS but turned off

Note: This example represents a possible Ascent or Entry CRT/DEU distribution.

B. DK Transmitter Enable Lts - GPC 4 ENA DK 1 - ON
GPC 4 ENA DK 3 - ON
all other lts - OFF

BFS XMTR ENA WD 1

BITS 1-6	BIT 7	BIT 8	BIT 9	BIT 10	BIT 11-16
	0	0	1	0	

BFC CRT Switch Lts - BFS DK SEL A - ON
BFS DK SEL B - ON
DEU ON/STBY Lts - DEU 1 ON - ON
DEU 3 ON - ON
all other lts - OFF

Analysis: CRT/DEU 1 - assigned to GPC 4
CRT/DEU 2 - unassigned and turned OFF
CRT/DEU 3 - assigned to BFS via BFC CRT switches
CRT/DEU 4 - unassigned and turned OFF

Note: This example illustrates a possible on - orbit CRT/DEU distribution. However, if GPC 4 (i.e. SM) Fails, there will be no CRT/DEU interface with the PASS. Undesirable procedures would be required to reestablish a PASS CRT/DEU interface.

4.6.8 Keyboard Inputs

When one of 32 available keys on the keyboard is depressed, two sets of codes are generated.

The first type of code consists of a four-digit hexadecimal "DEU key code." This H/W keycode is generated at the keyboard adapter level, stored in H/W register within the adapter, and transferred to the DCP (DEU S/W). There is one H/W keycode for each keyswitch depression.

The second type of code consists of a two-digit hexadecimal "keyswitch code". This code is generated by the DCP based on the four-digit code supplied via the keyboard adapater. The keycodes are reduced from four-digit to two-digit hexadecimal codes because of DCP size constraints and processing efficiency. These two-digit "keyswitch codes" are used by the GPC. They are packed as five-digit binary codes into words 3 to 12 of the DEU poll response word. (For information regarding troubleshooting of keycode problems, see OTP: DEU Memory Dump Applications, A.) The following table depicts the formation and association of both hardware and software keycodes.

The scratch pad line has a maximum number of characters that can be entered and an ERR will be annunciated if the maximum is exceeded (ref. paragraph 4.6.7.1). The maximum number of keyboard entries is typically 39; however, the number may be different depending upon the circumstances.

As a general rule, entries are counted as would normally be expected. Leading and separating spaces (denoted by '_') must be counted, however. The following is an example:

```
_ITEM_   = 6 spaces
123456
```

Each space, letter, number, decimal point, delimeter, and/or parenthesis is counted individually with blanks reserved for spacing at the beginning and end of the scratch pad line. The line is 51 spaces long, so spaces 1 and 51 are never filled with characters (fig. 4-31).

Since the SPL can never be overfilled by any entry except ITEM entries, the only concern is with the ITEM entries.

As stated earlier, characters are counted as might be expected. The exception is the delimeter key. By pressing a single key (+ or -), the delimeter key generates a blank, open parenthesis, Item number, close parenthesis, and sign (+ or -). The Item number appears in 2 spaces only and ranges from 00 to 99. An example is:

```
'_(14)+'
```

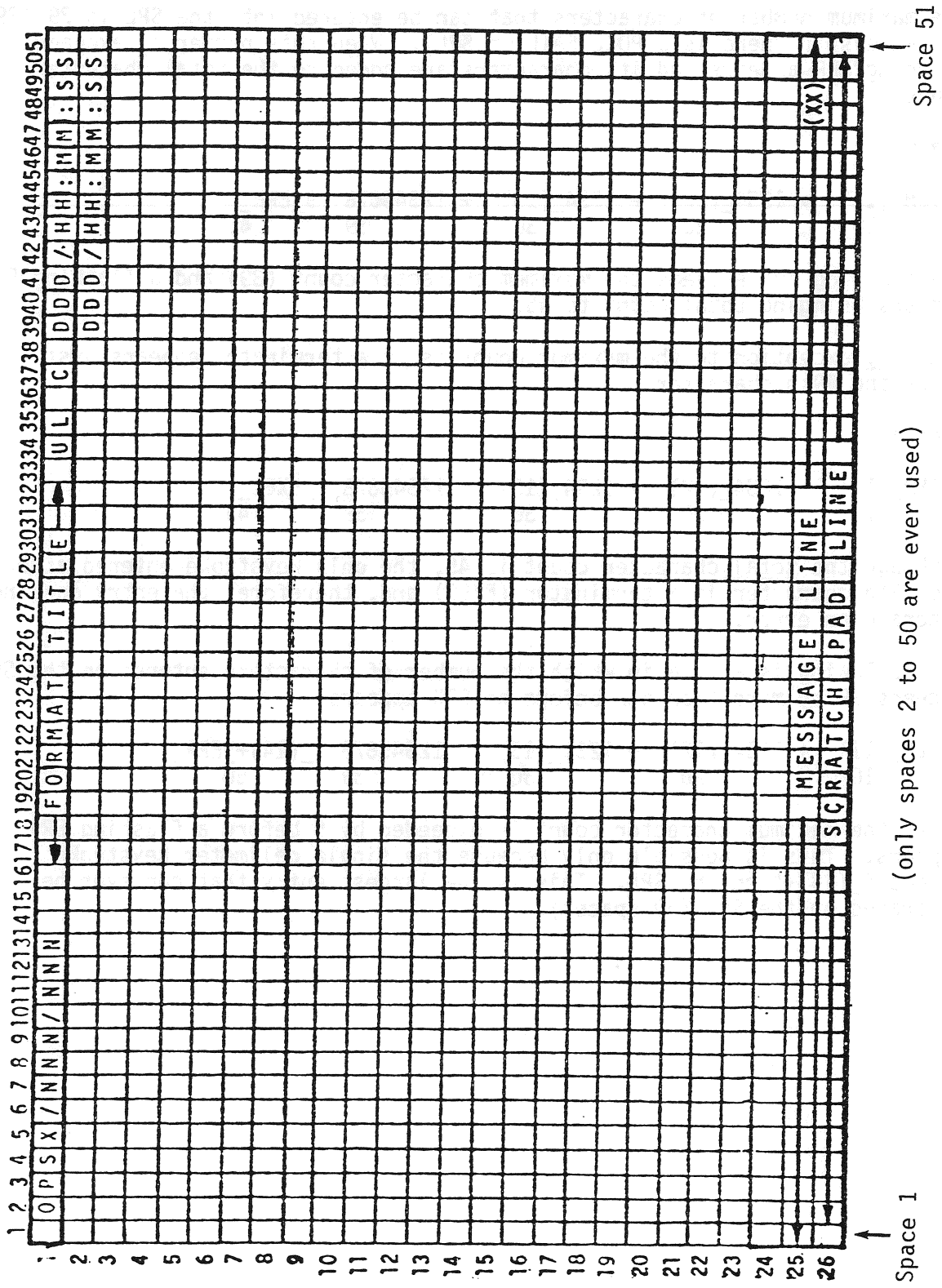


Figure 4-31.- CRT background format (Illus. 238).

Reserved for Figure 4-31.- CRT background format (Illus 238).

The maximum number of characters that can be entered into the SPL is 39 (29 characters if there is a POLL FAIL on SPL). When this number is exceeded, a flashing ERR appears and its characters are added to the total character count.

Example:

```
_ITEM_(11)  + 1234_(12)  + 1234_(13)  + 12345678  9_ERR_
      10           20           30           39       45
```

The trailing "9" exceeds the maximum character count (39) and a flashing ERR appears bringing total count to 45.

The only exception to the maximum count is if a terminator appears just after the 39th character.

Example

```
_ITEM_(11)  + 1234_(12)  + 1234_(13)  + 12345678  _EXEC_
      10           20           30           39       45
```

Although the total character count is 45, the only keystroke entered after the 39th character is a terminator (EXEC) and, therefore, the entry does not generate an error.

The following is a case in which the number of characters entered on the SPL exceeds 39 by more than one before an ERR appears.

```
_ITEM_(11)  + 1234_(12)  + 1234_(13)  + 12345678  _(14)+_ERR_
      10           20           30           39       50
```

Here, the maximum character count is exceeded by 6 before a flashing ERR appears. This is possible only because the single delimeter keystroke places "_(14)+" on the SPL. This is the largest entry that can ever be generated on the SPL (50 spaces).

TABLE 4-V.- DEU KEYSWITCH CODES

Switch	Keyswitch	DEU KYBD adapter keycode	DEU/GPC keycode
1	ACK	FFF9	1C*
2	MSG RESET	FFF1	1D*
3	SYS SUMM	FFE9	10
4	FAULT SUMM	FFE1	13
5	C	FFD9	0C
6	B	FFD1	0B
7	A	FFC9	0A
8	GPC/CRT	FFC1	19
9	F	FFFA	0F
10	E	FFBA	0E
11	D	FF7A	0D
12	I/O RESET	FF3A	18
13	3	FEFA	03
14	2	FEBA	02
15	1	FE7A	01
16	ITEM	FE3A	14
17	6	FFFB	06
18	5	FDFB	05
19	4	FBFB	04
20	EXEC	F9FB	1E
21	9	F7FB	09
22	8	F5FB	08
23	7	F3FB	07
24	OPS	F1FB	11
25	+	FFFC	16
26	0	EFFC	00
27	-	DFFC	15
28	SPEC	CFFC	12
29	PRO	BFFC	1F
30	.	AFFC	17
31	CLEAR	9FFC	1A
32	RESUME	8FFC	1B

*Keyswitch codes for ACK or MSG RESET are not packed in poll response words 3 to 12 for transmission to the GPC. Instead, the following bits are set in the poll response headerword:

- Bit 13 MSG RESET
- Bit 19 ACK

When either of these bits is set by the DEU, bit 21 "KYBD MSG present" flag of the headerword is not set.

4.6.9 Keyboard Syntax

The keyboard unit allows crew inputs to be transmitted to the GPC via the DEU. All keystrokes are displayed on the CRT SPL (line 26, bottom line of CRT) with the exception of the ACK, MSG RESET, and CLEAR keys.

The DEU performs a legal syntax check on each of the keystrokes entered on a keyboard. Figure 4-32 identifies the legal syntax flow. When the DEU identifies an invalid keystroke, it places a flashing ERR to the right of the illegal keystroke on the SPL (see MCDS Messages: DEU Annunciated Messages). The flashing ERR can be removed either by the depression of the CLEAR key, which will remove the ERR and the illegal keystroke, or by reinitiating a keystroke sequence. Note that the DEU checks only the keystroke syntax validity and not the legality of the keyboard message within an OPS. This check is made by the GPC, and a class 5 fault message, ILLEGAL ENTRY, is annunciated if the keyboard message is determined to be illegal (see MCDS Messages: GPC Annunciated Messages).

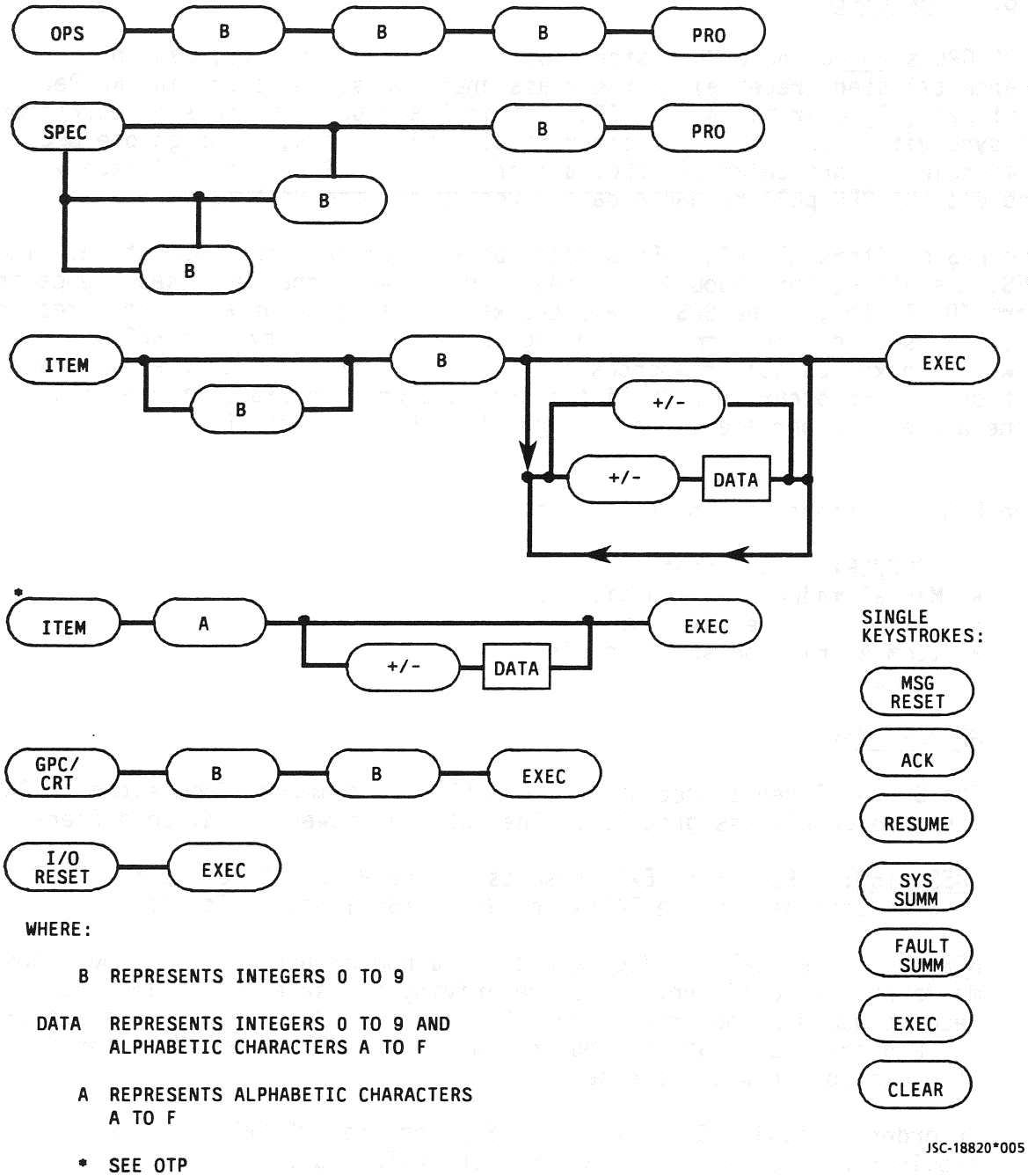


Figure 4-32.- Legal syntax flow.

4.6.10 DK Listen

PASS GPC's do not have DK listen capability and, therefore, can only transmit/listen (receive) to their assigned DEU's. However, in GNC Redundant Set (RS) operations, all GPC's "think" and perform the same operations in sync with each other. In order to accomplish this, although one GPC is commanding a particular CRT/DEU, all of the other RS GPC's will receive and process the DEU poll response data directly off the DK bus.

The BFS monitors all PASS DEU's continuously for certain information. The BFS, therefore, continuously "listens" on PASS-assigned DK buses, hence the term "DK listen." The BFS always checks the validity of a DEU poll response from DEU's under his command. During DK listen activity, the BFS also always checks DEU poll responses for validity. Should a single I/O transaction failure occur on a DK listen transaction, the BFS software will generate an appropriate error message (I/O ERROR CRT or CRT BITE).

The BFS DK listens for the following:

- "GPC/CRT" KYBD inputs
- Manual major-mode transitions
- "EXEC" (single key inputs)
- Item entries on specific displays
- Format index

A. GPC/CRT Key

The BFS will react according to GPC/CRT key commands from either a PASS-assigned or BFS-assigned DEU. The outcome, however, may be different.

PASS KYBD: GPC/CRT 52 EXEC results in the PASS GPC releasing DEU 2 (if it were assigned to the PASS) and GPC 5 (BFS) would pick it up.

BFS KYBD: GPC/CRT 52 EXEC results in a phenomenon known as "dual commanders" (two different GPC's are driving the same CRT). The GPC/CRT request would be honored by the BFS which would subsequently start driving a display on that CRT, but because the PASS has no DK listen mode, it would not know to release DEU 2.

In order to assign CRT X to the BFS using the GPC/CRT key, the user should first deassign the CRT from the PASS via GPC/CRT OX EXEC on a PASS keyboard. Then the assignment to the BFS via GPC/CRT 5X EXEC could be made to either BFS keyboard or a PASS keyboard (since BFS DK listens).

B. Manual Major-Mode Transitions

The BFS will react to the following manual major mode transitions:

OPS 1 :	102 → 103 or 601	OPS 6:	601 → 602
	103 → 104 or 601		602 → 603
	104 → 105		
	105 → 106		
OPS 3 :	301 → 302		
	302 → 303		
	303 → 304		
	304 → 305		

The BFS will perform the appropriate mode transition when it hears these input to either PASS or BFS keyboards. Note that the BFS will not react to OPS transitions.

C. EXEC (Single Key Inputs)

The BFS will listen and react to single key EXEC inputs to the maneuver display in the following OPS:

OPS 104
OPS 105
OPS 302

D. DK Listen Item Entries

The BFS will listen and react to certain item entries on specific displays as defined below:

DK listen items (OI-8B)

Display	Item no.	Description
HORIZ SIT	Items 3,4,5	Runway select
	Item 6	TAEM TGT (G&N)
	Item 7	MEP/NEP select
	Item 8	Aim pt. (headwind HI/LO)
	Item 9	Altimeter setting
	Items 40,41	TAL site select
XXXX MNVR YYYY	Items 1,2,3	OMS eng sel
	Item 4	RCS sel
	Items 6,7,8	TRIM loads
	Item 9	Vehicle weight
	Items 10-13	TIG (D,H,M,S)
	Items 14-18	PEG 4 TGTS
	Items 19-21	PEG 7 TGTS
	Item 22	'LOAD' CMD
	Item 23	Start CRT timer
	Items 24-26	Burn attitude data
	Items 28-33	GMBL select
	Item 35	Abort TGT/crew index
	"EXEC" CMD	Burn execute cmd
ENTRY TRAJ 1(2,3,4,5)	Item 1	Drag reference bias
OVERRIDE	Item 1	Abort TAL/AOA
	Item 2	Abort ATO
	Item 3	Abort
	Item 4	Max throttle
	Item 5	Crossfeed interconnect
	Items 6-8	Propellant dump commands
	Items 9-12	Centaur dump commands
	Items 13-14	Aft RCS
	Items 15-16	Forward RCS
	Items 17-18	Eleven select
	Items 20-21	Filter select
	Items 22-24	Atmospheric model select

E. Format Index

In order for the BFS to react to the "EXEC" and item entries to specific displays (paragraphs C and D above), it must be aware of which displays are being viewed on PASS CRT's. To accomplish this, the second word in each "Poll Response" includes a format index (table 4-II) which identifies to the BFS which critical format display is being viewed on the CRT. The format index is a cross-reference to the displays stored in the DEU critical format buffer (see OTP: DEU Memory Dump Applications, paragraphs A and B).

4.6.11 CRT Assignments

There are four user actions which affect DEU/CRT assignments.

- CRT MF switch changes
- GPC/CRT keyboard entries
- OPS transitions and OPS MODE recalls
- BFC CRT DISPLAY/SELECT switch selections

In order to have a thorough understanding of DEU/CRT assignments, the applicable DEU/CRT distribution data flow must be described. Figure 4-33 details the flow of data utilized in DEU/CRT reassignments. For example, both the CRT MF switch positions and the DEU/CRT distribution related keyboard entries (GPC/CRT XY EXEC and OPS XXX PRO) are transferred from the DEU to the GPC via the 16-word poll response over the DK buses. The BFC CRT select discretes, however, are routed to each GPC via the Backup Flight Controllers. Once a GPC has received the CRT MF switch position or the DEU/ CRT related keyboard entries, it shares that data and other pertinent data with common set GPC's via ICC. Note that the BFS does not receive or transmit over the ICC buses. However, the BFS is aware of the DEU/CRT's being commanded by PASS GPC's. The BFS turns on its receivers for the appropriate DK buses so that it can listen for certain information from the PASS DEU's including GPC/CRT keyboard entries (sec. 4.6.10A - COMMON DK LISTEN).

The PASS GPC's do not have a DK listen mode. A PASS GPC can only receive data from a DEU when it is actively commanding the DEU (except in the case where one redundant set GPC is commanding a DEU and all redundant set GPC's are receiving the response (reference paragraph 4.6.10). As mentioned earlier, the PASS GPC's depend primarily upon ICC traffic to relay DEU/CRT distribution data to other common set GPC's.

Included in this system brief are flow charts which functionally describe the DEU/CRT distribution following:

- A CRT MF switch change, figure 4-34
- A GPC/CRT keyboard entry, figure 4-35
- A memory configuation change, figure 4-36

Note that the PASS and BFS software which process the DEU/CRT assignments are system level; therefore, DEU/CRT distribution logic detailed by the flow charts is not OPS dependent. The flow charts are cross-referenced to a series of examples included at the end of this writeup. These charts attempt to describe realistic DEU/CRT reassignments but do not attempt to cover every possible combination of events leading to a particular DEU/CRT assignment. The following is a general discussion of CRT MF switch change logic, GPC/CRT keyboard entries, and memory configuration changes to supplement the flow charts. Also included is a description of how the BFC CRT DISPLAY/SELECT switches function and how they affect the other DEU/CRT distribution logic.

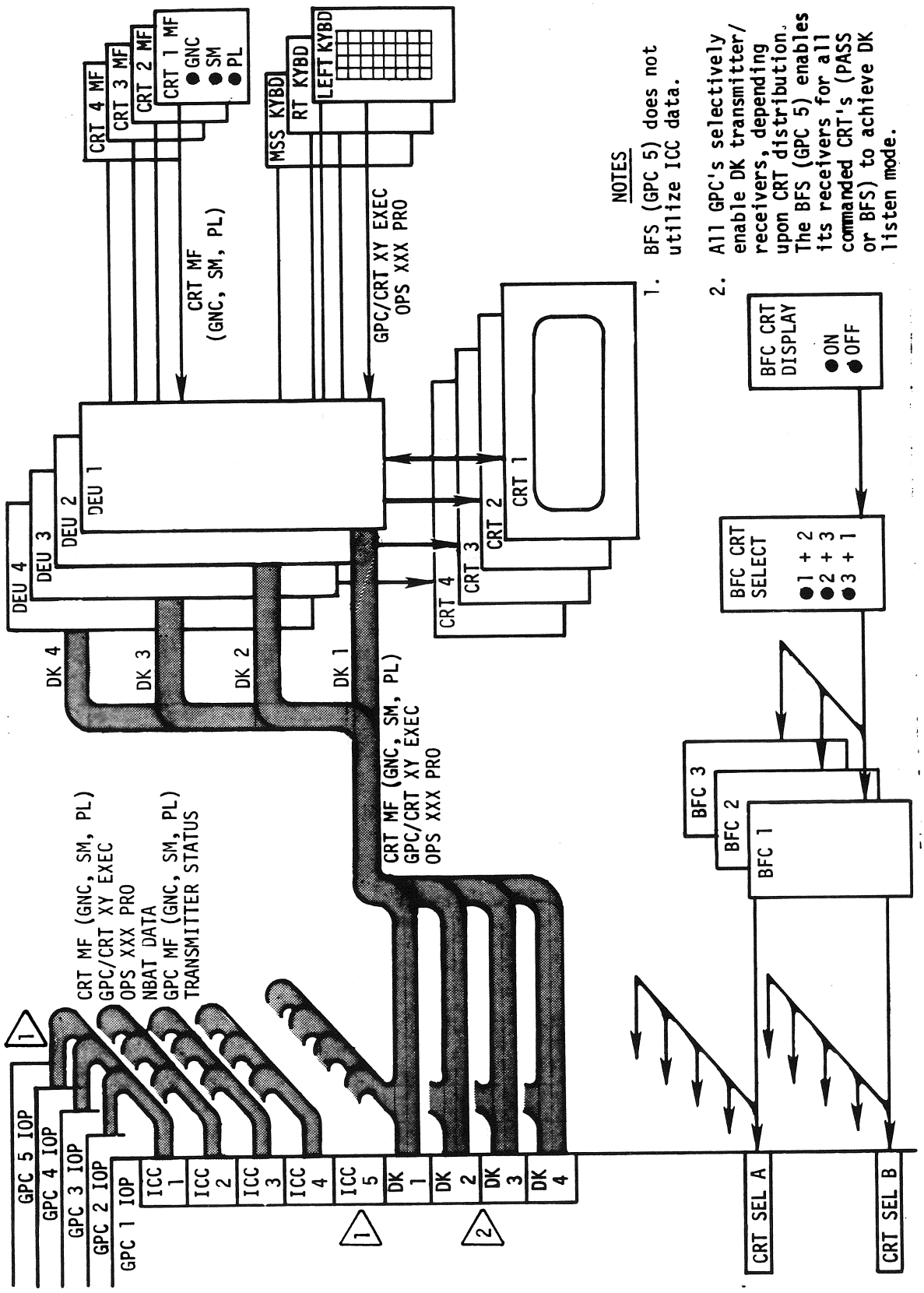
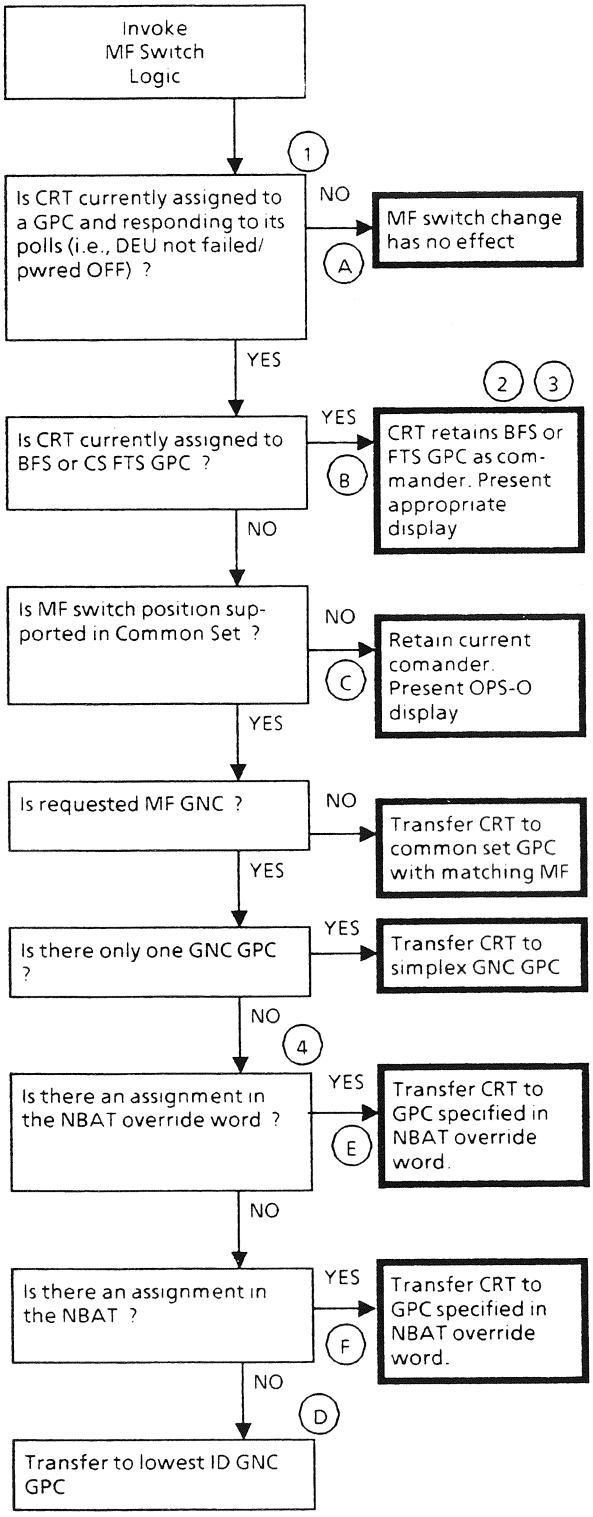


Figure 4-33.- Data flow for CRT assignments.

4.6.11.1 CRT MF Switch Logic

CRT MF switch logic (fig. 4-34) is invoked whenever a change in the CRT MF switch position is detected by a GPC, whenever a CRT reassignment is made via a GPC/CRT keyboard entry, or whenever a memory configuration change occurs. The CRT MF switch logic, therefore, appears on all three of the DEU/CRT distribution flow charts. The CRT MF switch logic is used to transfer a DEU/CRT within the common set. The logic first checks the NBAT override table (HAL name CZ2V-GPC-CRT-CMDR) to determine if there is an identified GPC commander. If not, the logic utilizes the Nominal Bus Assignment Table (NBAT) to determine the appropriate GPC with a matching MF for CRT transfer. Note that when the CRT MF switch is placed in a position that is not supported by any GPC's in the common set, the OPS-0 (or GPC Memory) display will be presented on the CRT. Also note that because of CRT MF switch logic, the CRT MF must be placed in an unsupported position for an OPS-0 GPC to retain the CRT. However, placing the CRT MF switch in an unsupported position will not automatically transfer a CRT to an OPS-0 GPC. The actual transfer must be accomplished via a GPC/CRT keyboard entry.

A form of the CRT MF switch logic is also invoked when a CRT assignment is made to the BFS or an FTS GPC. However, because neither the BFS not FTS GPC participates in ICC data transfer, CRT MF switch processing will not result in a DEU/CRT transfer. Instead, the logic will determine which display will be presented on the CRT.



(1) The CRT poll response is required because the MF switch position is part of the response

(2) CS FTS is a fail-to-sync PASS GPC which is operating outside of the Common Set

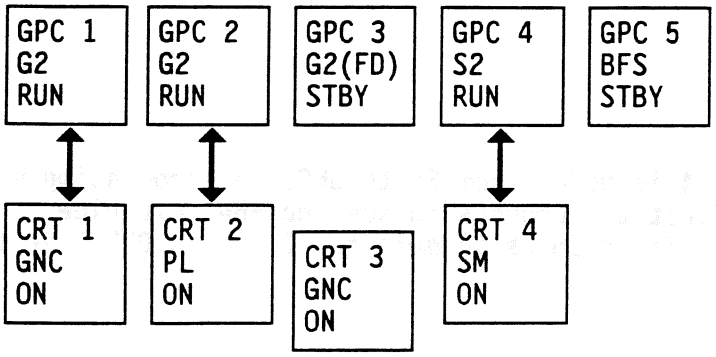
(3) The active OPS display will be presented if the CRT's MF switch position matches the MF of the PASS GPC. If there is no match, the GPC MEMORY display will be presented. The BFS supports both GNC and SM MF

(4) NBAT override word reflects the last assignment made via a GPC/CRT key since an OPS transition/OPS mode recall. Its HAL name is CZ2V_GPC_CRT_CMDR

Reserved for Figure 4-34.- The CRT major function switch.

CRT Major Function Switch Logic Examples

Each example presumes the following configuration



*NBAT:

CRT	G2	S2
1	1	4
2	2	4
3	0	0
4	0	4

NBAT OVERRIDE:

1	1	4
2	2	4
3	0	0
4	0	4

*Note: In the examples, only the CRT assignments for the active memory configurations are shown.

(A) If the MF switch for CRT 3 is moded from GNC to SM to PL, there will be no effect because there is not a GPC polling the CRT.

(B) If GPC 1 has failed to sync and is operating outside of the common set, moding the MF switch of CRT 1 to SM will not result in CRT 1 being transferred to GPC 4. Because GPC 1 has failed to sync, it cannot communicate the request from CRT 1 for SM via ICC data. Instead, GPC 1 will retain CRT 1 but will force the OPS-0 display on CRT 1 because it does not support the SM major function.

(C) CRT 2 is requesting the PL major function. Because PL is not supported in the common set, GPC 2 will retain the CRT and force the OPS-0 display on CRT 2.

(D) If the MF switch for CRT 4 is moded to GNC, GPC 4 will communicate that request to GPC's 1 and 2 via ICC data. Because CRT 4 does not have an NBAT override or NBAT assignment in G2, is will be transferred to the lowest numbered GPC (i.e., GPC 1) and CRT 4 will present the GNC OPS display.

(E) If the NBAT override is changed to:

CRT	G2	S2
1	1	4
2	2	4
3	0	0
4	2	4

And the MF switch for CRT 4 is moded to GNC, GPC 4 will communicate that request to GPC's 1 and 2 via ICC data. Because CRT 4 is assigned to GPC 2 in the G2 entry of the NBAT override, it will be transferred to GPC 2 and the GNC OPS display will be presented.

(F) If the NBAT override is as described initially (not as in sample E) and the NBAT is changed to:

CRT	G2	S2
1	1	4
2	2	4
3	0	0
4	1	4

and the MF switch for CRT 4 is moded from SM to GNC, the processing will assign CRT 4 to GPC 1. First the processing sees no NBAT override assignment for CRT 4, then finds an assignment for CRT 4 to GPC 1 in the NBAT.

4.6.11.2 GPC/CRT Keyboard Entries

The GPC/CRT keyboard entry logic (fig. 4-35) is invoked via a GPC/CRT XY EXEC keyboard entry made to a DEU/CRT that is being polled by a GPC. The flow chart differentiates between keyboard entries made to a DEU/CRT commanded by a common set GPC, the BFS GPC, and an FTS (PASS GPC that has failed-to-sync and is operating outside of the common set) GPC. The distinction is that a common set GPC has ICC data, but no DK listen; the BFS GPC has no ICC data, but has DK listen; and an FTS GPC has neither ICC nor DK listen data. The significance of the ICC data is that a GPC/CRT keyboard input is distributed to the common set via ICC. Note that CRT assignments to PASS GPCs via GPC/CRT keyboard entries result in NBAT override updates as specified by the flow charts. Note also that following a DEU/CRT assignment to a common set GPC, CRT MF switch logic is invoked.

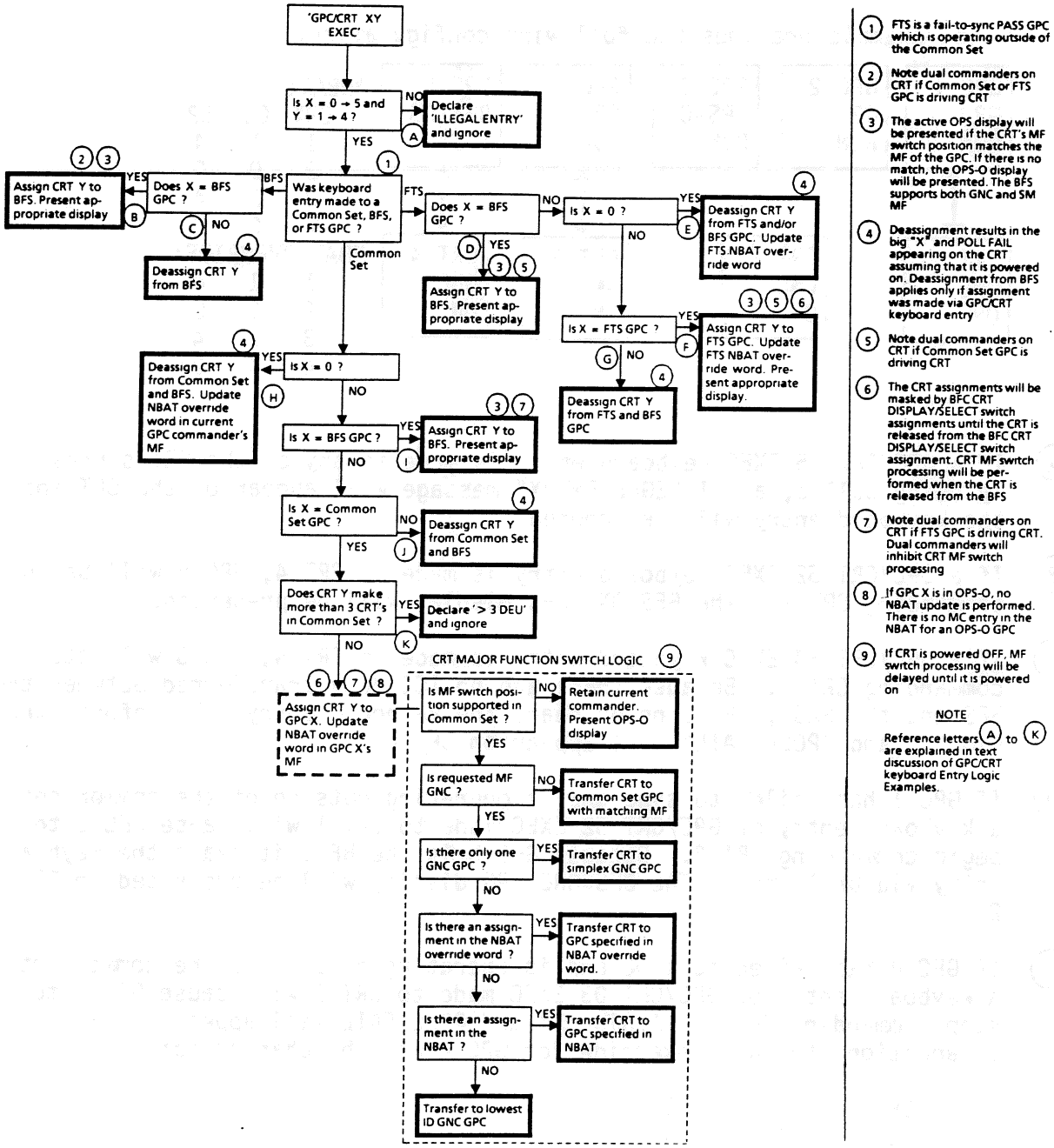
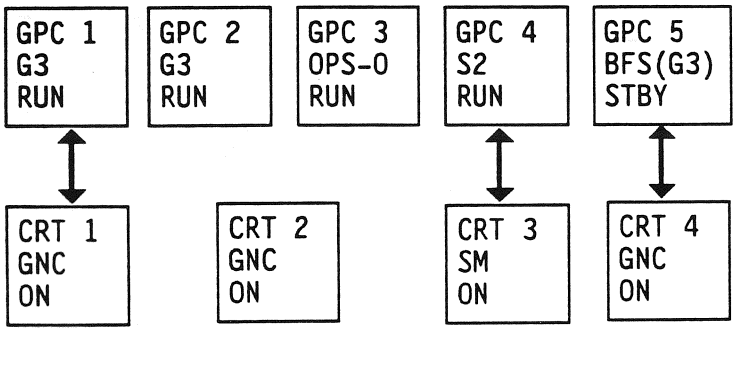


Figure 4-35.- The GPC/CRT keyboard entry.

GPC/CRT Keyboard Entry Logic Examples

Each example presumes the following configuration.



*NBAT:

CRT	G3	S2
1	1	4
2	0	0
3	2	4
4	0	0

NBAT OVERRIDE:

1	1	4
2	0	0
3	2	4
4	0	0

(A) If a GPC/CRT 25 EXEC keyboard entry is made to any of the CRT's presently being polled, an "ILLEGAL ENTRY" message will appear on the CRT and the keyboard entry will be ignored.

(B) If a GPC/CRT 52 EXEC keyboard entry is made to CRT 4, GPC 5 will begin commanding CRT 2. The BFS GNC OPS display will be presented.

(C) If a GPC/CRT 34 EXEC keyboard entry is made to CRT 4, GPC 5 will stop commanding CRT 4. Because there are no ICC data transferred between the BFS and the PASS, GPC 3 never hears the keyboard entry. Therefore, the big "X" and "POLL FAIL" will appear on CRT 4.

(D) If GPC 1 has failed to sync and is operating outside of the common set, a keyboard entry of GPC/CRT 52 EXEC made to CRT 1 will cause GPC 5 to begin commanding CRT 2. Because GPC 5 is the BFS, it hears the keyboard entry via DK listen. The BFS GNC OPS display will be presented on CRT 2.

(E) If GPC 4 has failed to sync and is operating outside of the common set, a keyboard entry of GPC/CRT 03 EXEC made to CRT 3 will cause GPC 4 to stop commanding CRT 3. A big "X" and POLL FAIL will appear on CRT 3. In addition, the NBAT override for GPC 4 will be changed to:

CRT	G3	S2
1	1	4
2	0	0
3	0	0
4	0	0

(F) If GPC 1 has failed to sync and is operating outside of the common set, a keyboard entry of GPC/CRT 12 EXEC made to CRT 1 will cause GPC 1 to begin commanding CRT 2. The GNC OPS display will be presented on the CRT. The NBAT override for GPC 1 will be changed to:

CRT	G3	S2
1	1	4
2	1	0
3	2	4
4	0	0

(G) If GPC 4 has failed to sync and is operating outside of the common set, a keyboard entry of GPC/CRT 13 EXEC made to CRT 3 will cause GPC 4 to stop commanding CRT 3. A big "X" and POLL FAIL will appear on CRT 3. Because there are no ICC data being exchanged between GPC 4 and the common set GPC's (i.e., GPC 1, 2, and 3), GPC 1 does not hear the keyboard entry.

(H) If a GPC/CRT 01 EXEC keyboard entry is made to any of the CRT's presently being commanded by common set GPC's (i.e., CRT's 1 and 3), GPC 1 will stop commanding CRT 1, and the G3 NBAT override will be updated as follows:

CRT	G3	S2
1	0	4
2	0	0
3	2	4
4	0	0

A big "X" and POLL FAIL will appear on CRT 1.

(I) If a GPC/CRT 51 EXEC keyboard entry is made to CRT 3, GPC 1 will stop commanding the CRT and GPC 5 will begin commanding it. GPC 1 heard the keyboard entry via ICC data and GPC 5 heard it via BFS DK listen. The BFS GNC OPS display will be presented on CRT 1. This transition does not affect the NBAT override.

(J) If GPC 4 has failed to sync and is operating outside of the common set, a GPC/CRT 41 EXEC keyboard entry made to CRT 1 will cause GPC 1 to stop polling CRT 1. However, because there are no ICC data being exchanged between GPC 4 and the common set GPC's, GPC 4 never hears the keyboard entry and does not initiate commanding of CRT 1. Therefore, the big "X" and POLL FAIL appear on CRT 1. This transition does not affect the NBAT override.

(K) If a GPC/CRT 14 EXEC keyboard entry is made to CRT 3, GPC 5 will stop commanding CRT 4 (keyboard entry is heard via DK listen) and GPC 1 will be commanding CRT 4 (keyboard entry is heard via ICC data). At this time there will be exactly three CRT's in the common set, which is the maximum allowed.

The NBAT override will will be:

CRT	G3	S2
1	1	4
2	0	0
3	2	4
4	1	0

If a GPC/CRT 22 EXEC keyboard entry is then made to CRT 3, a >3 DEU error message will appear on the common set CRT's and the keyboard assignment will be ignored. The CRT 2 will remain deassigned from the common set and the NBAT override will not be affected.

4.6.11.3 Memory Configuration Change

The memory configuration change logic (fig. 4-36) is invoked in a PASS GPC whenever an OPS transition or OPS mode recall is performed. During an OPS transition or OPS mode recall, the PASS system utilizes the software resident NBAT to determine required DEU/ CRT distribution. Note that the BFS does not utilize an NBAT and does not alter its DEU/CRT assignments during OPS transitions. The NBAT is a distribution table which includes assignments for target GPC's, strings, CRT's, etc., for all memory configurations. An example of the NBAT is provided in table 4-VI. The table is I-loaded (initial-loaded) into system software, but can be altered via item entries to the GPC MEMORY display. The CRT assignments via GPC/CRT or MF operations do not affect the NBAT (see discussions of NBAT override in section 4.5.11.2). Also, the NBAT override table is cleared when there is a memory configuration change.

The CRT MF switch position of a DEU/CRT is considered during a memory configuration change and has a direct affect on the CRT assignments. As shown by the flow chart, the CRT MF switch position will determine whether or not the NBAT assignment is performed.

Note that during memory configuration changes, a DEU/CRT which was being commanded by a common set GPC prior to the transition, but is not assigned per the new NBAT, will be retained after the transition. However, an NBAT assignment for a DEU/CRT which was not assigned to a common set GPC prior to the transition will be ignored.

A special transition not considered in the flow chart is when a PASS OPS-0 GPC is created by moding the GPC's Mode switch from STANDBY to RUN or from RUN to STANDBY to RUN. When a GPC is moded from RUN to STANDBY, all CRT assignments are released. When it is moded to RUN, the GPC will have no CRT's in the OPS-0 state unless there are no CRT's assigned to the current common set. In this case, it will begin polling CRT's 1, 2, and 3 with the exception that it will still honor BFC CRT DISPLAY/SELECT switch assignments. If a PASS OPS-0 GPC is created via an OPS-000 PRO keyboard entry, CRT assignments are not altered.

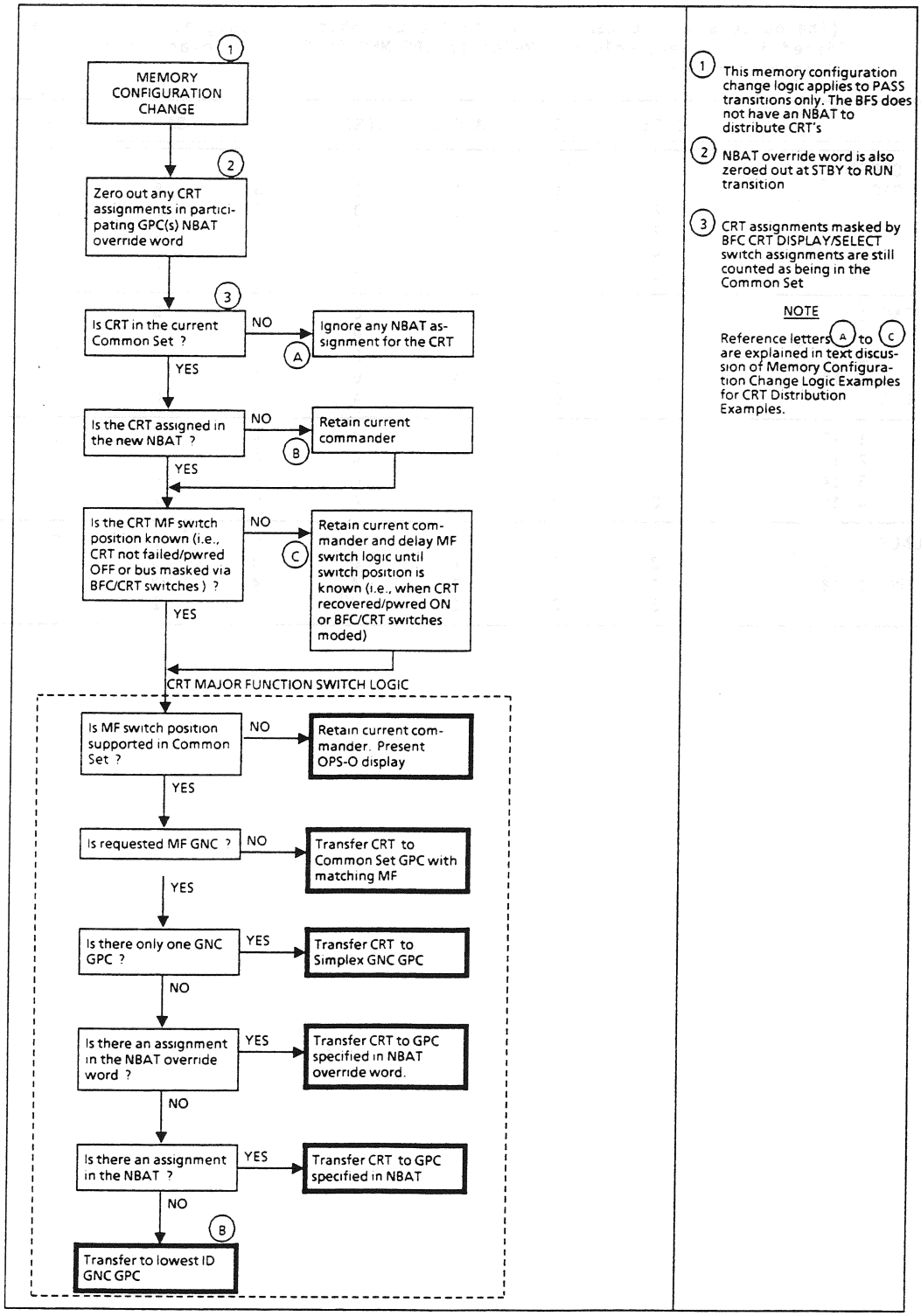


Figure 4-36.- Memory configuration change logic for CRT distribution (OPS transitions or OPS mode recalls).

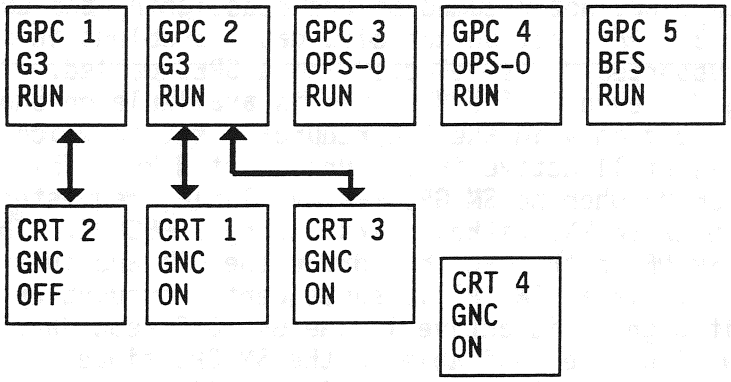
TABLE 4-VI.- NOMINAL BUS ASSIGNMENT TABLE (I-LOAD)

(The numbers in the table refer to the GPC which is assigned to the items listed in the left column. Refer to GPC MEMORY SPEC for clarification of item entries)

	1(G1)	2(G2)	3(G3)	4(S2)	5(S4)	6(P9)	8(G8)	9(G9)
1. CONFIG _ _(_ _)								
2. GPC _	1	1	1	0	0	0	1	1
_	2	2	2	0	0	2	2	2
_	3	0	3	0	0	0	0	3
_	4	0	4	4	0	0	0	4
_	0	0	0	0	0	0	0	0
STRING 1 7 _	1	1	1	0	0	0	1	1
2 8 _	2	2	2	0	0	0	2	2
3 9 _	3	2	3	0	0	0	2	3
4 10 _	4	1	4	0	0	0	1	4
PL 1/2 11	1	0	1	4	4	0	0	1
CRT 1 12 _	1	1	1	4	4	2	1	1
2 13 _	2	2	2	4	4	2	2	2
3 14 _	3	0	3	0	4	0	0	3
4 15	0	2	0	4	0	2	2	0
LAUNCH 1 16 _	1	0	0	4	0	0	0	1
2 17 _	2	0	0	0	0	0	0	2
MM 1 18 _	1	1	1	4	4	2	1	1
2 19	2	2	2	4	4	2	2	2

MEMORY CONFIGURATION CHANGE LOGIC FOR CRT DISTRIBUTION EXAMPLES

Each example presume the following configuration followed by GNC "OPS 301 PRO" keyboard entry to CRT 3.



NBAT:

CRT	G3
1	1
2	2
3	0
4	2

G3 Target GPC:

1	1
2	2
3	3
4	4
5	0

A Because CRT 4 is not in the current common set, it is not assigned to GPC 2 following the transition.

B Because CRT 3 is not assigned by the NBAT, GPC 2 will retain it. Note, however that MF processing will result in CRT 3 being transferred to GPC 1.

C Because CRT 2 is turned OFF, its MF switch position is not known. Therefore, the CRT 2 remains assigned to GPC 1. If CRT 2 is later turned ON, it will be transferred to GPC 2 via MF switch processing.

4.6.11.4 SPEC/ROLL-IN DISP Restrictions

Presently system software supports only two SPECs per major function (MF) and/or two ROLL-IN DISPs per MF. There is no restriction on the number of non-ROLL-IN DISPs, but procedurally, we never allow all CRTs to be actively driven from one MF; therefore only two non-ROLL-IN DISPs per MF will be active. Any attempt to support greater than two SPECs/ROLL-IN DISPs per MF will result in an "ILLEGAL ENTRY" message displayed on the CRT from which the request was made.

System SPECs are unique in that they can be called from any MF, but a particular system SPEC can be called in only one MF at a time in a single GPC or redundant set (see User Note 25396). Note that this does not preclude the crew from calling up the same system SPEC on two different CRTs that are commanded by differenct GPC(s) in different MFs (e.g., one GPC in GNC, commanding CRT 1 and one GPC in SM, commanding CRT 2; the crew can call up the same system SPEC on both CRTs).

If the crew calls the DPS UTILITY in the SM MF in one GPC or in a redundant set, the crew cannot call it up in the GNC MF in the same GPC/redundant set. A SPEC/DISP remains active on a CRT until a RESUME is entered on that CRT,

until an OPS transition, until OPS cleanup is done, or until a new SPEC/DISP is requested.

Moving the MF switch for a CRT terminates a DISP, but not SPECs or OPS displays (see level A par. 5.3.1.4.g and 1). This action can cause problems if a spec was called on a CRT, but not resumed before deassigning the CRT, as in postinsertion for CRT 3. This action can also cause problems in a GPC/redundant set if an unsupported MF is selected and a SPEC called. If not resumed before changing MF, even if the MF later is available on another GPC, the SPEC will still be hung only in the GPC/redundant set in which it was called since that SPEC is still active in the unsupported MF. This can happen if you selected the SM MF when no SM GPC was available, requested a SPEC, and then took the MF back to GNC without first doing a RESUME. The SPEC remains active in the SM MF in the GPC that drove the CRT and in any GPCs in redundant set with it. If an SM GPC is subsequently brought up and the same SPEC requested, although it is active in the GNC GPC/redundant set under the SM MF, the SPEC will not be available on the SM GPC since it is not active in the SM GPC. The SPEC will not be available if requested in the GNC MF in the GMC GPC(s), since it is active under the SM MF, not the GNC MF. If the MF switch in the GNC GPC is switched to the SM MF, the SM GPC will take control of that CRT and the SPEC will remain active in the GNC GPC (a request to call up the SPEC under the GNC MF will be rejected).

There are three approaches to clearing out such a hung system SPEC in the GNC GPC/redundant set example discussed above.

A. Pro the SM GPC to OPS 0 or mode it to STBY, change the MF switch of the CRT where the SPEC is loaded to SM and depress the RESUME key.

B. Perform an OPS transition or an OPS mode recall for the set with the hung SPEC with a target set change (expansion or contraction).

C. Mode the GNC GPC(s) to STBY.

The last two approaches will perform an OPS cleanup that will clear out all the CRT displays including the SM MF assignment.

The ROLL-IN DISP capability is presently implemented only for the SM MF. With this capability, eight SM DISPs and all the CREW TEXT DISPs can be called.

When these restrictions and certain CRT/GPC failure cases are integrated together, the viewing capabilities of the system depend on how software has interpreted the failure. Anytime the GPC/CRT interface fails, the software does not cancel the active SPEC/DISP, but still counts the "lost" SPEC/DISP against its MF limitations. In other words if the crew had been viewing a SPEC or a ROLL-IN DISP on a specific CRT and that CRT now has an "I/O ERROR" or "BITE" FSP message annunciated against it, the S/W assumes that the SPEC/ROLL-IN DISP is still active. As a matter of fact the "lost" SPEC/ROLL-IN DISP can be called up on another CRT; however, the S/W will not release the SPEC/ROLL-IN DISP from the failed CRT.

To workaround these restrictions, several recovery techniques are available.

A. At any OPS transition involving a memory configuration change; i.e., G2→G3, all SPECs DISPs, and ROLL-IN DISPs are canceled. As a subset to this, if any SPEC is major mode dependent then proceeding to another major mode will also cause a cancellation request to be issued; e.g., MM 301 supports SPEC 21 and 22, but MMs 302 and subsequent do not.

B. It is possible for the crew or ground to make S/W think that it heard the RESUME key input from the failed CRT to recover a "lost" SPEC. The crew procedure involves a READ/WRITE technique, whereas the ground can uplink a resume request. Since the crew did not believe the R/W technique was acceptable, the ground will uplink a resume request via DEU EQUIVALENT providing all the following conditions are met:

   1. Ground has AOS and uplink capability to the GPCs.

   2. Affected DK bus must be assigned to the active PASS GPC from which the SPEC/ROLL-IN DISP was called.

   3. The uplink command must exactly duplicate the failed CRTs MF S/W setting that is seen on downlist.

C. To remove the limitation of two ROLL-IN DISPs being active after a GPC/CRT interface failure, the ground using DEU EQUIVALENTS which obey the same commanding criteria as above, will uplink a non-roll-in DISP request to the failed CRT and then uplink a resume request.

   During the active entry phase, four specs (GPC MEM, RCS, HORIZ SIT, and OVERRIDE) and zero ROLL-IN DISPs are available to the crew. Potentially, we could trap one of these specs and thus be restricted to only one additional spec. Normally, the crew uses the HORIZ SIT and OVERRIDE SPECs, but uses the other two in failure cases. Thus, if we trap either the HORIZ SIT or OVERRIDE SPEC, the ground should do

This page intentionally blank

nothing. If GPC MEM or the RCS SPEC is trapped, the ground should uplink a resume to the failed CRT.

4.6.11.5 BFC CRT DISPLAY/SELECT Switch Selections

The BFC CRT DISPLAY and SELECT switches (located on panel C3) are the primary method used for assigning DEU/CRT's 1, 2, and 3 to the BFS (GPC/CRT keyboard entries are the secondary method). Table 4-VII summarizes the combination of DISPLAY and SELECT switch positions versus BFS CRT assignments for both BFS preengaged and engaged. Note that during a GPC IPL, the CRT upon which the IPL MENU will be displayed is determined by the BFC CRT switches.

TABLE 4-VII.- BFC CRT DISPLAY/SELECT SWITCH LOGIC

BFC CRT		BFC CRT discretes		BFS CRT assignment[a]	
Display	Select	CRT SEL A	CRT SEL B	Pre-engaged	Engaged
OFF	N/A	0	0	0	1 and 2
ON	1 + 2	0	1	1	1 and 2
ON	2 + 3	1	0	2	2 and 3
ON	3 + 1	1	1	3	3 and 1

[a]Primary CRT assignment is relinquished if the IPL SOURCE switch is in the MM1 or MM2 position (i.e., CRT 3 will be relinquished by the BFS (big X, POLL FAIL) if the BFC/CRT switches are ON and 3 + 1 for either PRE-ENGAGED or ENGAGED). Assignment is reestablished when IPL SOURCE switch is moded to OFF.

Therefore, the BFS will relinquish control of the appropriate CRT when the IPL SOURCE switch is moded to the MM1 or MM2 position. The discretes produced by the BFC CRT switches are hardwired to all five GPC's. Therefore, all GPC's (common set, BFS, and FTS) will react to a BFS CRT selection via BFC CRT switch positions. If a GPC is commanding a DEU/CRT and detects BFC CRT discretes assigning that DEU/CRT to the BFS, it will leave its transmitter on for that DEU but will set a bus mask to prevent dual commanders when the BFS begins commanding the DEU/CRT. By leaving its DK transmitter on, the GPC will automatically command the DEU/CRT when the DEU/CRT is released from the BFS via the loss of the appropriate BFC CRT discretes. Therefore, the flow charts for the CRT MF switch logic, the GPC/CRT keyboard entries, and the memory configuration changes apply regardless of whether or not a DEU/CRT is assigned via BFC CRT discretes to the BFS; except, as noted on the flow charts, certain processing may be delayed until a DEU/CRT is released from the BFS. Also, because a PASS GPC leaves its transmitters on for a DEU/CRT selected by BFC CRT discretes, the DEU/CRT will still be

counted in the common set when the ">3 " logic is invoked. A series of examples is provided at the end of this procedure to show how the CRT distribution flow charts are affected by BFC CRT assignments.

Because the BFS depends upon a single set of two discretes (select A and select B) to determine CRT assignments via the select switch, it is susceptible to single discrete BFS IOP failures which affect CRT assignments. The PASS redundancy manages these two select discretes across the common set and are therefore less sensitive to single discrete failures at a particular PASS IOP. When the BFS GPC incurs a failure of either select A or select B at its IOP, often dual commanders between the PASS GPCS and the BFS GPC of a CRT results. Note that this will only happen if the failure is at the BFS IOP, and not at the select switch itself. Failures of the select switch are sensed at all GPCs (PASS and BFS) and there is no disagreement on CRT assignments between the PASS and BFS. Table 4-VII summarizes the effects of select A or select B discrete failures at the BFS IOP. The table assumes the BFS is ON and in STBY or RUN and that CRTs 1, 2 and 3 are assigned to the PASS.

The PASS redundancy management of the CRT select discretes needs further explanation. The redundancy management is done on the CRT number represented by the select A and B discretes rather than on the individual discretes. This means that if a given PASS GPC has an input discrete failure such that it senses discrete A on, then for redundancy management purposes that GPC sees CRT 2 selected. If a second PASS GPC senses both discretes A and B set, then that GPC sees CRT 3 selected. In a three-GPC set, this is significant. In the case stated above, two out of the three GPCs in the set are seeing false CRT select discretes; but, because there is no majority agreement on the CRT selected, the redundancy managed CRT selection remains as it did before the discretes failed on at the two GPCs (CRT 0). If the discretes had been redundancy managed individually, then for the above example, there would be a majority of the GPCs (two of three) seeing discrete A on; and, therefore, the redundancy managed CRT selected would have been changed to CRT 2; and the PASS GPCs would have stopped polling CRT 2.

TABLE 4-VIII.- DISCRETE FAILURES AT BFS IOP

SWITCH POS		SELECT A		SELECT B	
DISP	SEL	FAILED ON	FAILED OFF	FAILED ON	FAILED OFF
ON	3 + 1	No effect	Big X/PF CRT 3 DC on CRT 1	No effect	Big X/PF CRT 3 DC on CRT 2
	1 + 2	Big X/PF CRT 1 DC on CRT 3	No effect	No effect	Big X/PF CRT 1
	2 + 3	No effect	Big X/PF CRT 2	Big X/PF CRT 2 DC on CRT 3	No effect
OFF	3 + 1	DC on CRT 2	No effect	DC on CRT 1	No effect
	1 + 2	DC on CRT 2	No effect	DC on CRT 1	No effect
	2 + 3	DC on CRT 2	No effect	DC on CRT 1	No effect

DC: Dual Commanders
PF: Poll Fail

BFC CRT DISPLAY/SELECT SWITCH LOGIC EXAMPLES

A. In example F of the GPC/CRT KEYBOARD ENTRY LOGIC EXAMPLES, if the BFC CRT DISPLAY switch is ON with the BFC CRT SELECT switch in the 2 + 3 position, the BFS GPC will be commanding CRT 2. The keyboard entry of GPC/CRT 12 EXEC will cause GPC 1 to enable its transmitter for DK 2. However, the BFC CRT discretes will cause GPC 1 to place a bus mask on DK 2. Therefore, the BFS will continue to command CRT 2 until the BFC CRT DISPLAY switch is moded to OFF or another position is chosen for the BFC CRT SELECT switch, at which time GPC 1 will command CRT 2 and the GNC OPS display will be presented. The NBAT override update for GPC 1 will occur while the BFS is still in command of CRT 2.

B. In example C of the MEMORY CONFIGURATION CHANGE LOGIC FOR CRT DISTRIBUTION EXAMPLES, consider that CRT 2 is ON and that prior to the transition, the BFC CRT DISPLAY switch is ON and the SELECT switch is in 2 + 3. Therefore, the MF switch position is unknown because although GPC 1 still has its transmitter on, its bus is masked. The MF switch logic will be delayed until the BFC/CRT DISPLAY switch is moded to OFF or the SELECT switch is moded to another position (3 + 1 or 1 + 2). At that time, CRT 2 will be transferred to GPC 2.

4.6.12 Uplink - DEU Equivalent

The ground can uplink keystrokes to the PASS GPCS via a DEU equivalent uplink capability whenever a poll flag is on in the GPC (the BFS does not have this capability). The CRT does not have to be powered on or assigned to perform this uplink, and the crew does not see the keystrokes on the CRT scratch pad line.

A. A GPC does not check whether or not a CRT is assigned to it.

B. A GPC does check the poll flag.

C. The poll flag is turned off for critical BITE bits only.

A GPC will process a DEU equivalent load if the poll flag is on regardless of whether or not the CRT is assigned.

Problems

A. If a DEU load is sent to the wrong MF but that MF GPC "thinks" a SPEC is on that CRT, the GPC will attempt to process.

B. If the load is sent emulating the wrong CRT, the GPC will attempt to process if it believes a SPEC is called.

C. If a CRT is de-assigned, the above two notes apply with the added problem of having no insight into that CRT.

Therefore, do not send DEU equivalent loads to a CRT that is not being "actively" polled by the GPC (MF) to which the load is addressed. The DEU equivalent keystrokes are stored in the same GPC memory location as the poll response keystrokes and can therefore be seen in the downlist. However, the next valid poll response from the DEU will overlay the DEU equivalent keystrokes. With a downlist rate of once per second and a DEU poll rate of twice per second, we often do not see the DEU equivalent keystrokes in the downlist. However, if the CRT is turned OFF (i.e., no valid DEU poll response), the DEU equivalent keystrokes will remain in the downlist. The PASS processing of the DEU equivalent keystrokes is slightly different than the processing of DEU poll response keystrokes up to a point. However, the keystrokes are repacked for processing of ITEM entries, SPEC calls, etc., by DMC - Super. The PASS no longer makes a distinction as to where the keystrokes come from (see software flows PASS MCDS INPUT PROCESSOR (drawing number 4.7.1) and PASS MCDS MESSAGE PROCESSOR (drawing number 4.7.2)).

The initial slight difference in processing of DEU equivalent keystrokes versus poll response keystrokes typically creates a problem for uplinks when a CRT has failed. The MCDS software forces an artificial major function switch processing wherever it sees a GPC/CRT reassignment of the CRT (see fig. 4-35).

In order to force MF switch processing, the software artificially sets a locally stored version (i.e., not downlisted) of the MF to a -1 whenever a GPC/CRT reassignment is made. This local version of MF is updated to a valid value upon receiving the MF from a DEU POLL response (DEU equivalents bypass this leg of the software and cannot update the local MF flag). DMC_SUPER checks the MF on all messages it processes (including DEU equivalents) and stops processing the message if the MF does not correspond to the local MF flag. Unfortunately, if the local MF flag is -1, our DEU equivalent will never be processed. This will happen if the crew performs a GPC/CRT assignment after a CRT has been turned off or failed (i.e., no valid

poll response). One of the ways we use DEU equivalents is to RESUME specs or ROLL-IN DISPS for failed CRT's. If the crew has performed trouble-shooting including a GPC/CRT reassignment, the uplinked RESUME will not work. However an OPS transition will clear out the SPECS and ROLL-IN DISPS. For GNC, a transition to OPS 8 or a target set charge will work. For SM, going to OPS 000 and back to 201 will work.

Not all keystrokes can be processed via DEU equivalent uplink. The key-strokes that cannot be processed are FAULT SUMM, SYS SUMM, MSG RESET, ACK, GPC/CRT, CLEAR. This restriction is built into the ground MOC computer during the build of the DEU equivalent uplink load. It is not a restriction of the onboard software.

4.6.13 Roll-in Display Processing

The following report is a description of R19 User Interface (UI) software processing required to generate a Roll-in display upon crew request through the Multifunction CRT Display System (MCDS). The data required to generate a roll-in display, such as a Crew Text or CRYO/FUEL CELL (Spec 68), is stored on Mass Memory Unit (MMU). IBM terms these displays "NON-CRITICAL" while displays stored in the DEU and in GPC memory are termed "CRITICAL".

Each display is defined by a Display Format Table (DFT). The DFT provides all the information to be used by the GPC and DEU to generate the display. The DFT is broken into two parts - the Dynamic Data Table (DDT) and Background Data. The DDT holds all information for all dynamic parts of the display (i.e. what parameters are used, the location of the parameter on the display, scaling factors, and the refresh rate). The Background Data table defines all non-dynamic portions of the display (i.e. titles, lines to be drawn and parameters that are not cyclically updated). When a display is requested the Background Data must be moved to the DEU for processing by the DEU Control Program (DCP) and the location of the DDT must be moved to MCDS Allocation Table (MAT). The The Cyclic Display Processor (DCICYC) which runs at a rate of 2Hz uses the data in the MAT to access the DFT. Each DDT entry associated with the DFT is processed and the Format Control Words required to update the display are generated. These FCWs are then written by DCICYC to the area of the DEU reserved for dynamics display FCWs.

Most displays (i.e. critical) DFTs are resident in GPC memory in the Display Format Buffers (DFB). There are a maximum of six DFBs accessible depending on GPC status. DFB1 which holds the DFTs for Spec 0, Fault Summ Page and the Roll-in Display Error Status page is always accessible. DFBs two and three which hold the remaining critical DFTs are not accessible during a GPC reconfiguration, these DFBs are activated once the GPC overlay is complete. DFBs four and five are only accessible in SM2 since they are used to temporarily hold the requested Roll-in display's DFT. DFB6 or the Display Mass Memory Directory identifies all non-critical displays as well as their location on the MMU except for the Crew Text displays (made only of Background data) whose MMU locations are numerically computed since they are of fixed size. Note that R19 UI does not allow ITEM parameters. This is to be change for non-critical SPACELAB SPECS in R20 User Interface.

Processing Logic:

Figure 4-37 is a simplified structured control flow for DMC_SUPER, the UI Operations control supervisor. On this flow, the highlighted (dotted) path shows the logic for a nominal roll-in display request. When DMC_SUPER sees a keyboard event, which is set by DMM_MCDS_PROCESSING, DM1_KEYBOARD (fig. 4-37) is called. DM1-KEYBOARD checks each DEU for processing requests. If the first keystroke is valid, DM1_KEYSTROKE PROC is called. Note here that ITEM, OPS, I/O RESET, and GPC/CRT will not be processed when a reconfiguration is in progress. After processing is done by DM1_KEYSTROKE_PROC and all modules it calls, polling is restored, under interrupt protection, to the DEU if necessary. If an error was found during processing, then the error annunciation processor FMPT_UI_OPERR is queued and the error (Illegal Entry) is logged.

Figure 4-39 shows the control flow for DM1_KEYSTROKE_PROC Spec key processing. If the request is not for a Spec page (i.e., a display page) DM3_Display (fig. 4-40) is called.

DM3 DISPLAY checks DFBs one through five for the DFT of the requested display. Critical displays will be found in DFB one, two, or three. A Roll-in display could be found in DFB4 or DFB5 if another DEU is using the display or if the display was previously requested then released and not written over by another display request. If the Roll-in display was found here then the display request is treated as if it were a critical display. If the display was not found and the configuration is SM2, DXR_DMM_ROLLIN (case 2) is called to locate and display the Roll-in display. If DXR_DMM_ROLLIN was unable to find or process the display, the error processor is queued for Illegal Entry and processing returns to DM1_KEYBOARD.

Figure 4-41 shows the logic for DXR_DMM_ROLLIN. Case 2 is used to locate the display and issue the MM-I/O request. If preparation for another non-critical display request is in progress and it is not the same as the previous request, then the error condition is set and processing returns to DM3_DISPLAY. If the present request is the same as the previous request, then the output tables of the previous request are updated to include the present DEU and processing returns to DM3_DISPLAY (see figure). If the request is a crew text request (i.e., page no. ≥ 900) then the MM load block is calculated. Otherwise DFB6 is searched to verify that the request is for a valid page. If the page was not found in DFB6, processing returns to DM3_DISPLAY with the Illegal Entry condition set. Otherwise the load block and size are recorded. Next DXR_BUF_TEST is called to find a free buffer.

Processing returns to DMM_ROLLIN with DFB four or five (four first) if the data in the buffer is not being used by a DEU. BUF_TEST will return with DFB four or five (four first) if they are being used by the requesting DEU. If the above Buffer tests fail, then processing returns to DM3_DISPLAY with the error condition Illegal Entry. Otherwise FIOSVC is called to schedule the I/O. Then DM4_DEU_PROC is called to update the MAT. Processing then returns to DM3_DISPLAY.

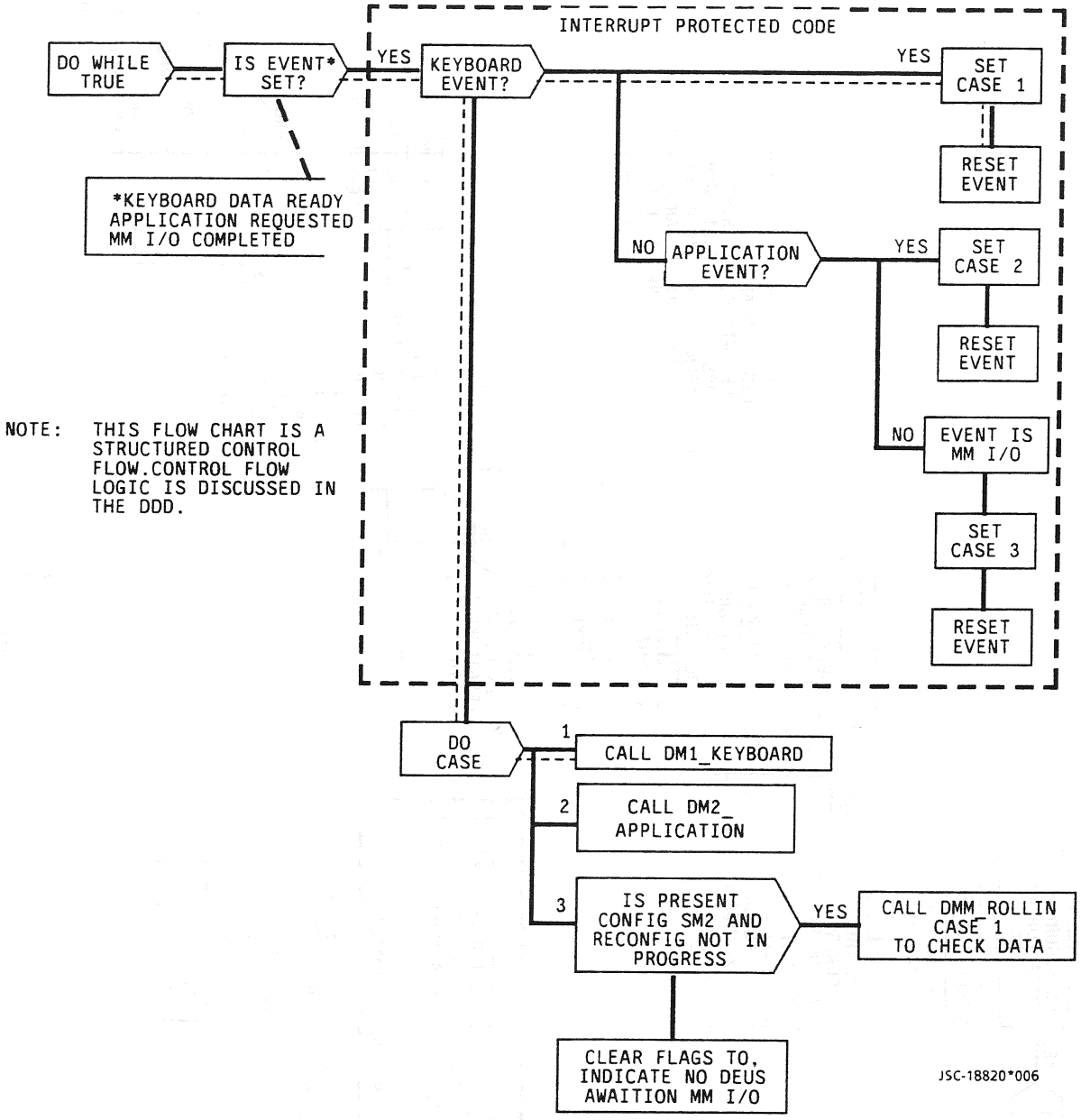


Figure 4-37.- DMC_SUPER.

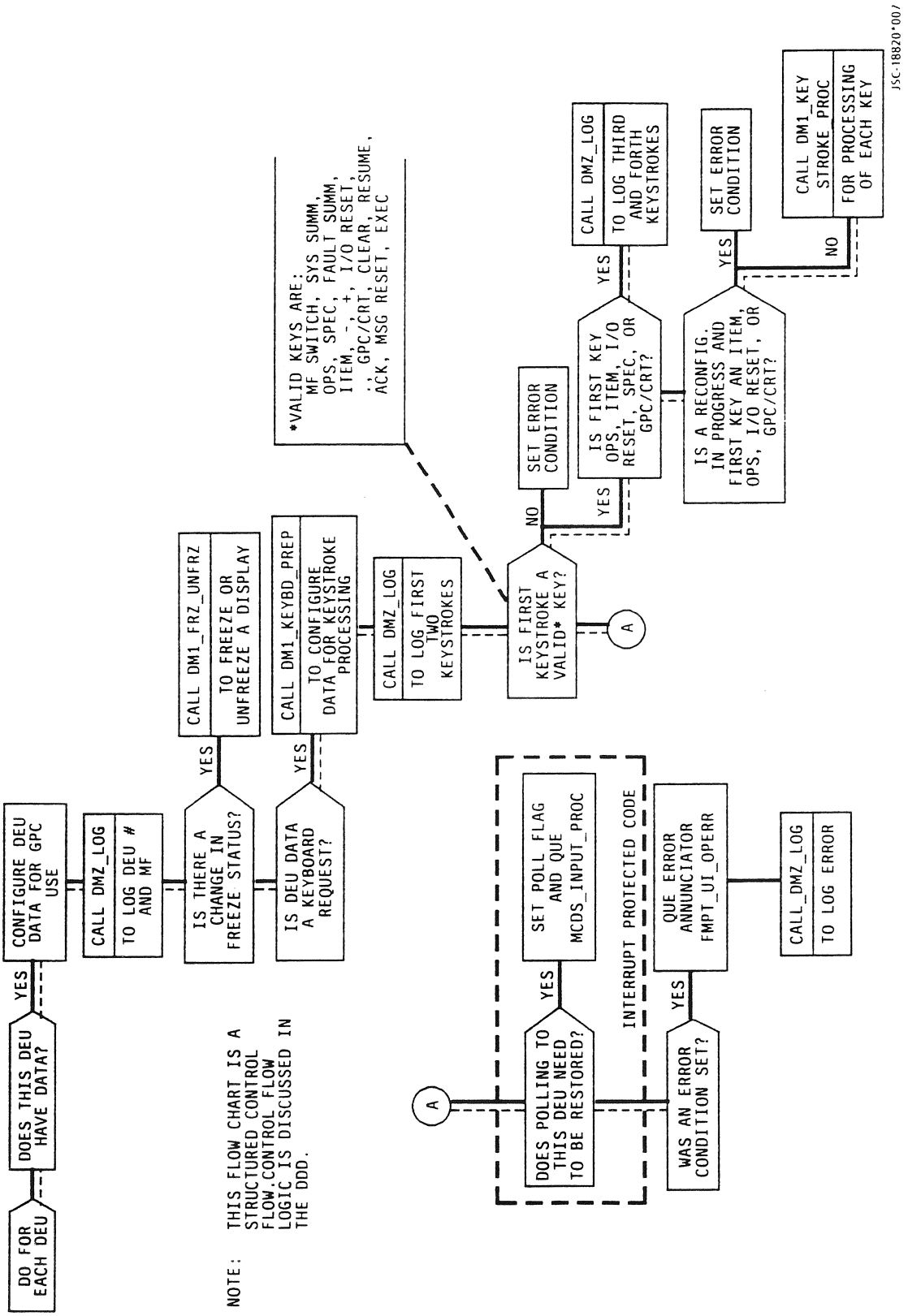


Figure 4-38.- DM1_KEYBOARD.

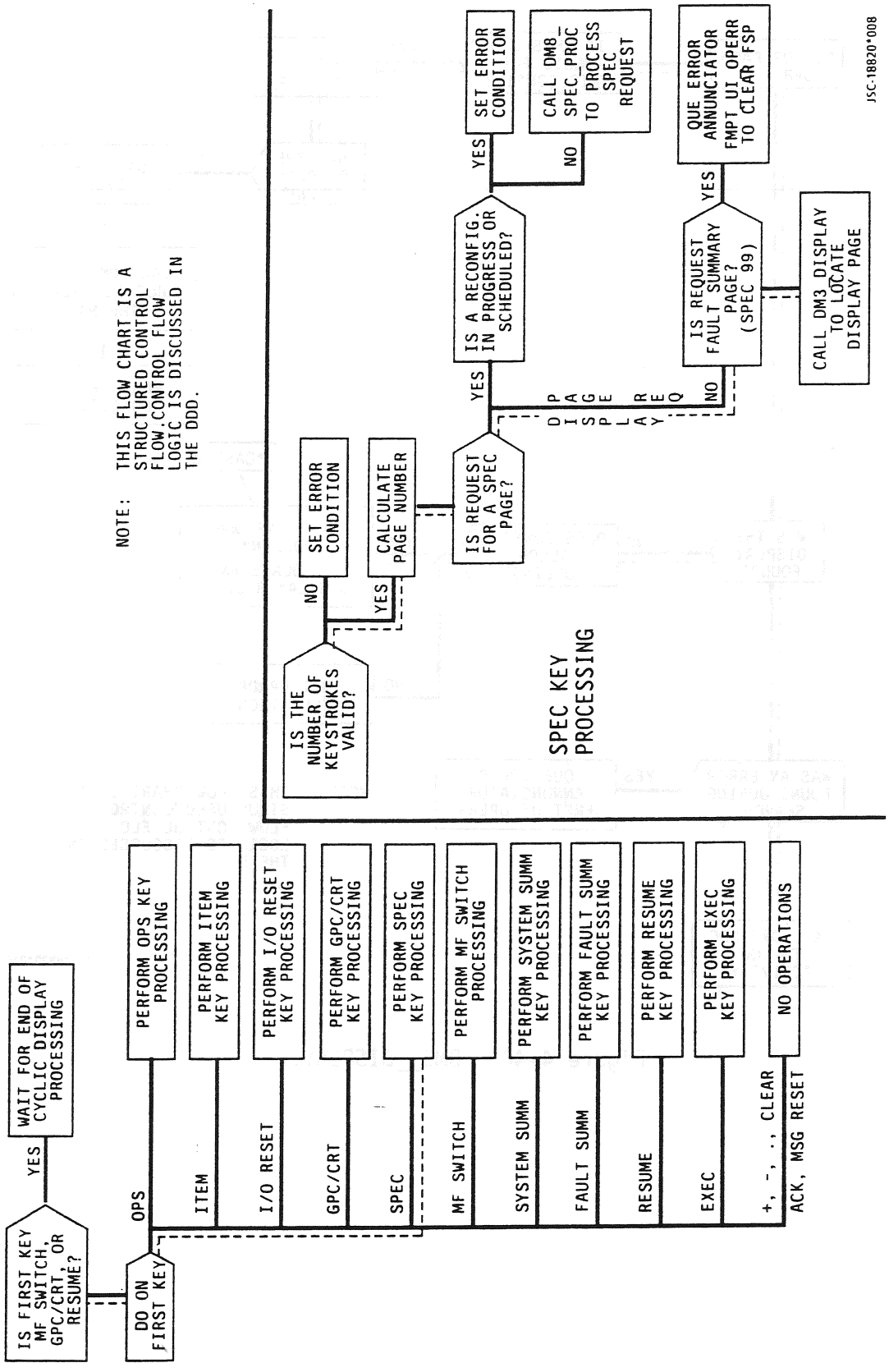


Figure 4-39.- DM1_KEYSTROKE_PROC.

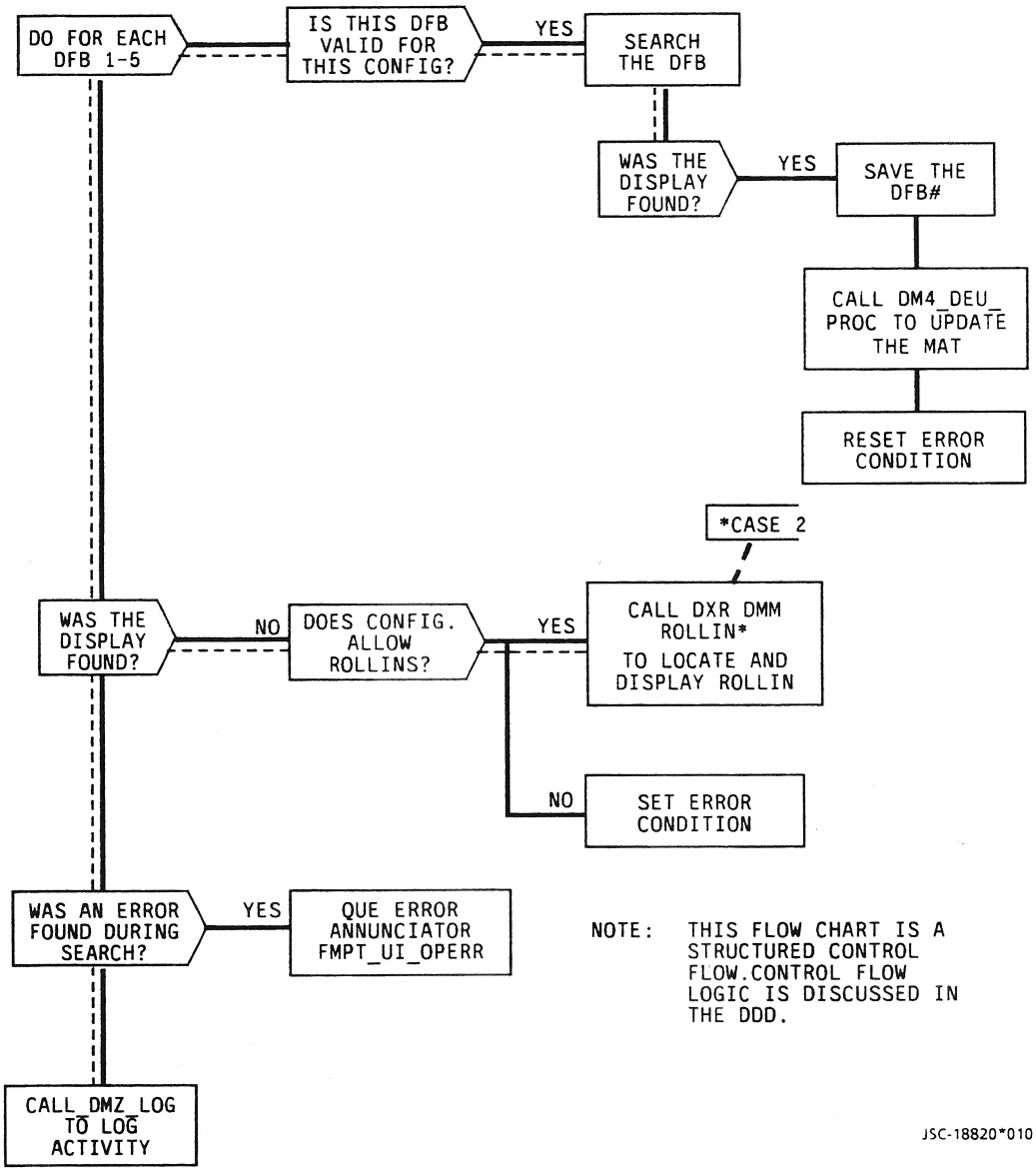


Figure 4-40.- DM3__DISPLAY.

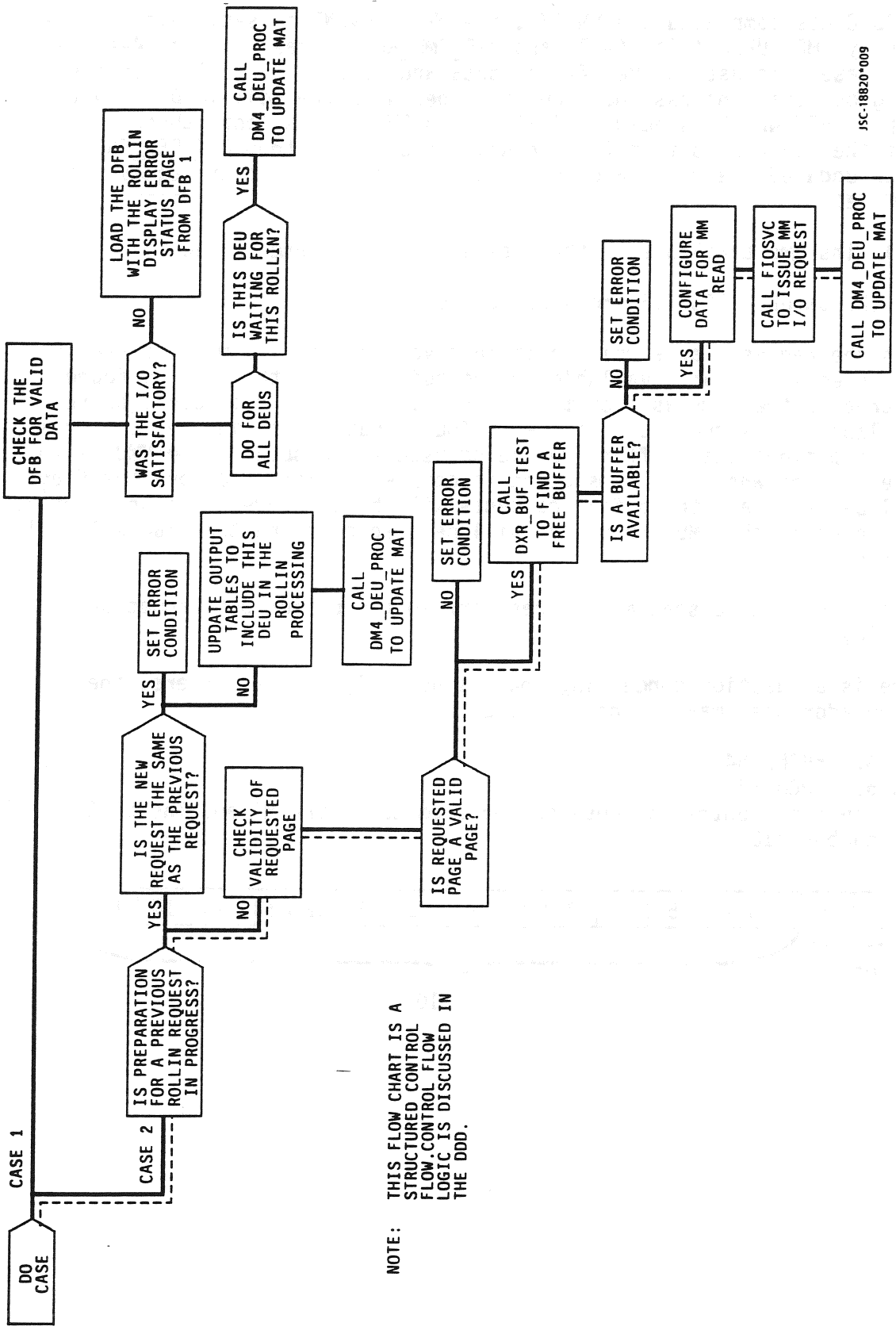


Figure 4-41.- DXR_DMM_ROLLIN.

Once FIOSVC has completed the MM I/O, the MM IO EVENT is set, which is de-detected by DMC_SUPER (fig. 4-37) and DXR_DMM_ROLLIN case 1 (fig. 4-41) is called. Case 1 is used to verify the data and update it to MAT. If the I/O was not good, then the assigned DFB is loaded with the Roll-in Display Error Status page DFT which is held in DFB1. Then DMM ROLLIN loops through each DEU. If the DEU was waiting for the Roll-in display, then DM4_DEU_PROC is called to updated the MAT. Processing of the Roll-in display request is then complete.

What roll-ins are available to the crew if both MMUs are failed?

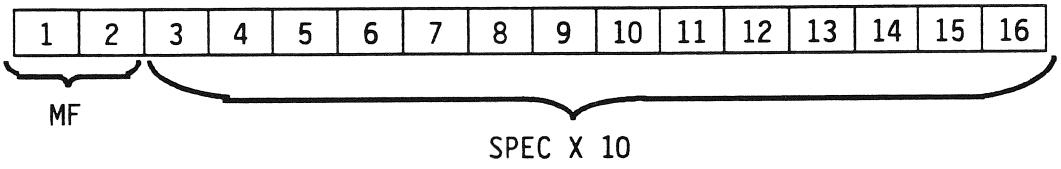
A. For the SM GPC, DFB 4,5 are used for roll-in specs.

B. When the request for a roll-in is received, a check is made to see if that spec is already available in the buffers. If the Spec is found, it is treated the same as a critical format. If it is not found, buffer availability is checked. If one is found that is not in use, a flag is set to prevent that buffer from being used for display. The MMU is then accessed and when the transaction is completed without error, the flag will be reset and the buffer contents will be displayed. If an error is encountered, the MMU Error page will be displayed and the flag will NOT be reset.

C. This means that a spec may be resident in the buffer but cannot be displayed.

If there is a question concerning what is actually in the buffers, the following addresses may be interrogated.

DFB 4...#PCDG154
DFB 5...#PCDH155
The tenth entry in these compools contains the MF and The SPEC number X10.



4.6.14 MCDS Parameters

4.6.14.1 D/L Parameters

MSID	Name	HAL-Name	No. bits
DEU power			
V73S2001E (OI)	DEU 1 - ON	N/A	1
V73S2011E (OI)	DEU 2 - ON	N/A	1
V73S2021E (OI)	DEU 3 - ON	N/A	1
V73S2051E (OI)	DEU 4 - ON	N/A	1
V73S2002E (OI)	DEU 1 - STBY	N/A	1
V73S2012E (OI)	DEU 2 - STBY	N/A	1
V73S2022E (OI)	DEU 3 - STBY	N/A	1
V73S2052E (OI)	DEU 4 - STBY	N/A	1
CRT MF switch			
V92U6717CX	CRT 1 MF (SWTICH)	TBS	2
V92U6717CY	CRT 1 MF (SWTICH)		2
V92U6776CX	CRT 2 MF (SWTICH)		2
V92U6776CY	CRT 2 MF (SWTICH)		2
V92U6840CX	CRT 3 MF (SWTICH)		2
V92U6840CY	CRT 3 MF (SWTICH)		2
V92U6900CX	CRT 4 MF (SWTICH)		2
V92U6900CY	CRT 4 MF (SWTICH)		2
DEU 1			
V98M3976P	DEU 1 HDR BUF	TBS	
V92M6721PX	DEU 1 RESPONSE WORD 1		
V92M6721PY	DEU 1 RESPONSE WORD 1		
V92M6721PZ	DEU 1 RESPONSE WORD 1		
V72M5650PX	DEU 1-MCDS STATUS WORD 1		
V72M5650PY	DEU 1-MCDS STATUS WORD 1		
V72M5650PZ	DEU 1-MCDS STATUS WORD 1		
V72M5670PX	DEU 1-MCDS STATUS WORD 2		
V72M5670PY	DEU 1-MCDS STATUS WORD 2		
V72M5670PZ	DEU 1-MCDS STATUS WORD 2		
V72M5680PX	DEU 1-MCDS STATUS WORD 3		
V72M5680PY	DEU 1-MCDS STATUS WORD 3		
V72M5670PZ	DEU 1-MCDS STATUS WORD 3		

MSID	Name	HAL-Name	No. bits
DEU 2			
V98M3977P	DEU 2 HDR BUF		TBS
V92M6780PX	DEU 2 RESPONSE WORD 1		
V92M6780PY	DEU 2 RESPONSE WORD 1		
V92M6780PZ	DEU 2 RESPONSE WORD 1		
V72M5720PX	DEU 2-MCDS STATUS WORD 1		
V72M5720PY	DEU 2-MCDS STATUS WORD 1		
V72M5720PZ	DEU 2-MCDS STATUS WORD 1		
V72M5740PX	DEU 2-MCDS STATUS WORD 2		
V72M5740PY	DEU 2-MCDS STATUS WORD 2		
V72M5740PZ	DEU 2-MCDS STATUS WORD 2		
V72M5750PX	DEU 2-MCDS STATUS WORD 3		
V72M5750PY	DEU 2-MCDS STATUS WORD 3		
V72M5750PZ	DEU 2-MCDS STATUS WORD 3		
DEU 3			
V98M3978P	DEU 3 HDR BUF		TBS
V92M6844PX	DEU 3 RESPONSE WORD 1		
V92M6844PY	DEU 3 RESPONSE WORD 1		
V92M6844PZ	DEU 3 RESPONSE WORD 1		
V72M5810PX	DEU 3-MCDS STATUS WORD 1		
V72M5810PY	DEU 3-MCDS STATUS WORD 1		
V72M5810PZ	DEU 3-MCDS STATUS WORD 1		
V72M5830PX	DEU 3-MCDS STATUS WORD 2		
V72M5830PY	DEU 3-MCDS STATUS WORD 2		
V72M5830PZ	DEU 3-MCDS STATUS WORD 2		
V72M5840PX	DEU 3-MCDS STATUS WORD 3		
V72M5840PY	DEU 3-MCDS STATUS WORD 3		
V72M5840PZ	DEU 3-MCDS STATUS WORD 3		
DEU 4			
V98M3979P	DEU 4 HDR BUF		TBS
V92M6904PX	DEU 4 RESPONSE WORD 1		
V92M6904PY	DEU 4 RESPONSE WORD 1		
V92M6904PZ	DEU 4 RESPONSE WORD 1		
V72M5910PX	DEU 4-MCDS STATUS WORD 1		
V72M5910PY	DEU 4-MCDS STATUS WORD 1		
V72M5910PZ	DEU 4-MCDS STATUS WORD 1		
V72M5930PX	DEU 4-MCDS STATUS WORD 2		
V72M5930PY	DEU 4-MCDS STATUS WORD 2		
V72M5930PZ	DEU 4-MCDS STATUS WORD 2		
V72M5940PX	DEU 4-MCDS STATUS WORD 3		
V72M5940PY	DEU 4-MCDS STATUS WORD 3		
V72M5940PZ	DEU 4-MCDS STATUS WORD 3		

MSID	Name	HAL-Name	No. bits
BFS MCDS			
DEU MSG BUFFER:			
V98M3960P	DEU MSG BUF WD 1	TBS	16
V98M3961P	DEU MSG BUF WD 2		16
V98M3962P	DEU MSG BUF WD 3		16
V98M3963P	DEU MSG BUF WD 4		16
V98M3964P	DEU MSG BUF WD 5		16
V98M3965P	DEU MSG BUF WD 6		16
V98M3966P	DEU MSG BUF WD 7		16
V98M3967P	DEU MSG BUF WD 8		16
V98M3968P	DEU MSG BUF WD 9		16
V98M3969P	DEU MSG BUF WD 10		16
V98M3970P	DEU MSG BUF WD 11		16
V98M3971P	DEU MSG BUF WD 12		16
V98M3972P	DEU MSG BUF WD 13		16
V98M3973P	DEU MSG BUF WD 14		16
V98M3974P	DEU MSG BUF WD 15		16
V98M3975P	DEU MSG BUF WD 16		16

Note: The DEU MSG buffer are the 16 words from any CRT poll response that contains a complete message (bit 13 of poll header word) or one that fails the checksum processing. See 4.6.1.2.2 for a description of these 16 words. This buffer will be updated for complete CRT messages for both CRT's assigned to the BFS and for CRT's heard via DK listen. The poll header word (DEU MSG BUF WD 1) will identify which CRT the message corresponds to. Because this buffer is overlaid when a complete message is detected, in periods of high keyboard activity, it is possible to "lose" the downlisted keyboard information because of the slow downlist rate (once per second). Also, it is possible that a downlist cycle will hit while the buffer is being updated. In this case, for one downlist cycle there will be a mix of part of the old 16 words with part of the new 16 words.

DEU HDR BUFFER:	
V98M3976P	DEU 1 POLL HDR WD
V98M3977P	DEU 2 POLL HDR WD
V98M3978P	DEU 3 POLL HDR WD
V98M3979P	DEU 4 POLL HDR WD

Note: This buffer contains the poll response header words from DEU's 1 to 4 and is updated for each DEU when a MSG RESET (bit 5), ACK (bit 11), or DISPLAY FREEZE (bit 12) bit is set or a MF switch position change is detected (bits 9 and 10). The buffer is updated from poll responses received by the BFS for both CRT's assigned to the BFS and for CRT's heard via DK listen.

DEU HDR BUFFER:	
V98M3980P	DEU ERR BUF HDR
V98M4864P	DEU ERR BUF H/W1
V98M4862P	DEU ERR BUF H/W2
V98M4863P	DEU ERR BUF S/W

Note: This buffer contains the poll header word and the three DEU BITE words from the last DEU POLL response in which an SASTP, CRITICAL BITE, or Initialization Required Bit is set (bits 14, 15, and 16) respectively of the poll header word) or when a valid DEU BITE bit is set (bits 1, 2, 3, 4, 5, 7, 8, 12, 15 of H/W 1; bits 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 of H/W2; bits 0, 2, 3, 4, 5, 7, 9, 11, 12, 13, 14, 15 of S/W). This buffer is updated for both CRT's assigned to the BFS and for CRT's heard via DK listen. The poll header word will identify which CRT incurred the error.

DEU CMDR/ILL ENT INDICATOR:

V98M3959P BFS Keyboard Transaction Monitor (Parent word)
V98H4396C - DEU 1 CMD STAT/ILL ENT CNT (bits 1 to 4)
V98H4397C - DEU 2 CMD STAT/ILL ENT CNT (bits 5 to 8)
V98H4398C - DEU 3 CMD STAT/ILL ENT CNT (bits 9 to 12)
V98H4399C - DEU 4 CMD STAT/ILL ENT CNT (bits 13 to 16)

Note: This downlisted parameter indicates the command status of each DEU, (whether commanded by PFS or BFS), and a count of illegal keyboard messages received from each DEU when commanded by either PFS or BFS. (Only PFS entries which are DK listen are checked for validity.) Each DEU is allocated four bits as identified above, and each group of four bits is interpreted as follows:

Binary	Hex	Definition
0000	0	PFS commanding, No illegal entries.
0001	1	PFS commanding, one illegal entry.
0010	2	PFS commanding, two illegal entries.
0011	3	PFS commanding, three illegal entires.
0100	4	BFS commanding, no illegal entries.
0101	5	BFS commanding, one illegal entry, last processed minor cycle 3.
0110	6	BFS commanding, two illegal entries, last processed minor cycle 3.
0111	7	BFS commanding, three illegal entries, last processed minor cycle 3.
1000	8	BFS commanding, one illegal entry, last processed minor cycle 11.
1001	9	BFS commanding, two illegal entries, last processed minor cycle 11.
1010	A	BFS commanding, three illegal entries, last processed minor cycle 11.
1011	B	BFS commanding, one illegal entry, last processed minor cycle 19.
1100	C	BFS commanding, two illegal entries, last processed minor cycle 19.
1101	D	BFS commanding, three illegal entries, last processed minor cycle 19.
1110	E	Not used.
1111	F	Not used.

General note:

NOTE: Downlist sends these buffers to the ground piecemeal. A new message will overwrite the message buffer sometime after downlist has sent part of the previous message. Since downlist will continue to send a previous complete message until a new message is received, this will definitely occur. This means that for that cycle the ground will receive a composite message - the first part of the previous message and the second part of the new message. This condition will, of course, last only until downlist again sends the entire buffer (about 1 second).

4.6.14.2 MCDS - MCC Special Comps

A more complete description of comps can be found in Console Handbook, apppendix C, and in Level B/C Telemetry Requirements For Shuttle (MOC), section 5.3.3.25):

DEU display

M29G2500L Display DEU 1
M30G2500L Display DEU 2
M31G2500L Display DEU 3
M32G2500L Display DEU 4

Major function

M25G2500R MF DEU 1
M26G2500R MF DEU 2
M27G2500R MF DEU 3
M28G2500R MF DEU 4

SPL

M01G2500R Scratch Pad 1 - Word 1
M02G2500R Scratch Pad 1 - Word 2
M03G2500R Scratch Pad 1 - Word 3
M04G2500R Scratch Pad 1 - Word 4
M05G2500R Scratch Pad 1 - Word 5
M06G2500R Scratch Pad 1 - Word 6

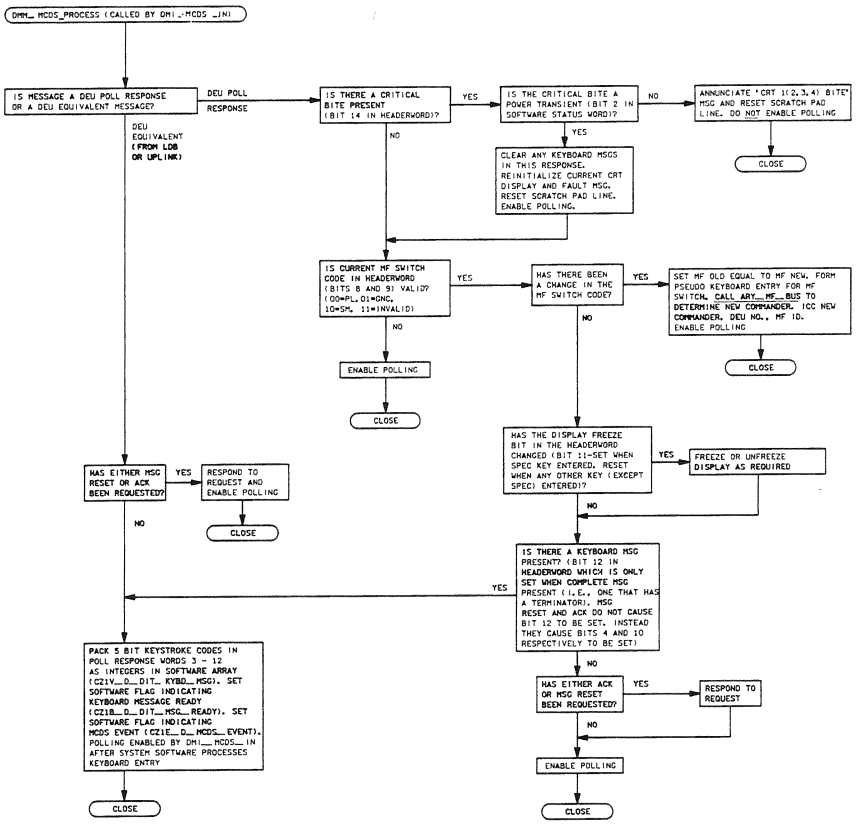
M07G2500R Scratch Pad 2 - Word 1
M08G2500R Scratch Pad 2 - Word 2
M09G2500R Scratch Pad 2 - Word 3
M10G2500R Scratch Pad 2 - Word 4
M11G2500R Scratch Pad 2 - Word 5
M12G2500R Scratch Pad 2 - Word 6

M13G2500R Scratch Pad 3 - Word 1
M14G2500R Scratch Pad 3 - Word 2
M15G2500R Scratch Pad 3 - Word 3
M16G2500R Scratch Pad 3 - Word 4

M17G2500R Scratch Pad 3 - Word 5
M18G2500R Scratch Pad 3 - Word 6

M19G2500R Scratch Pad 4 - Word 1
M20G2500R Scratch Pad 4 - Word 2
M21G2500R Scratch Pad 4 - Word 3
M22G2500R Scratch Pad 4 - Word 4
M23G2500R Scratch Pad 4 - Word 5
M24G2500R Scratch Pad 4 - Word 6

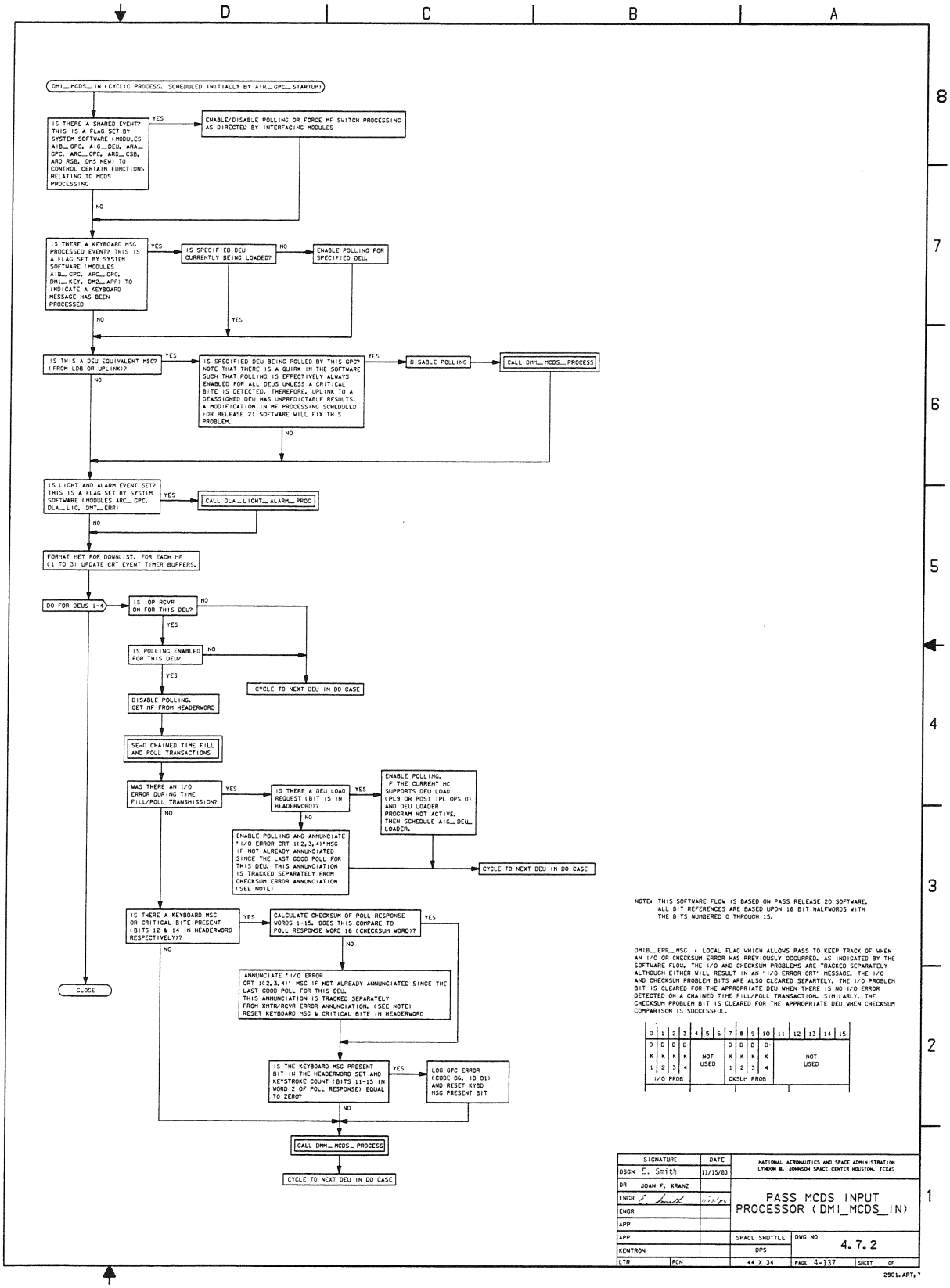
4.7 MCDS SOFTWARE FLOWS



NOTE: THIS SOFTWARE FLOW IS BASED ON PASS RELEASE 20 SOFTWARE. ALL BIT REFERENCES ARE BASED UPON 16 BIT HALFWORDS WITH THE BITS NUMBERED 0 THROUGH

SIGNATURE		DATE	NATIONAL AERONAUTICS AND SPACE ADMINISTRATION LYNDON B. JOHNSON SPACE CENTER HOUSTON, TEXAS
DSGN	E. Smith	11/15/83	
DR			
ENGR	E. Smith	11/17/83	PASS MCDS MESSAGE PROCESSOR (DMM_MCDS_PROCESS)
ENGR			
APP			
APP			SPACE SHUTTLE DWG NO 4.7.1
KENTRON			DPS
LTR	PCN		44 X 34 PAGE 4-135 SHEET 1 OF 1

2900.ART.7



SIGNATURE		DATE	NATIONAL AERONAUTICS AND SPACE ADMINISTRATION LYNDON B. JOHNSON SPACE CENTER HOUSTON, TEXAS
DSGN E. Smith		11/15/83	
DR JOAN F. KRANZ			PASS MCDS INPUT PROCESSOR (DM!_MCDS_IN)
ENGR E. Smith		11/15/83	
ENGR			
APP			
APP			SPACE SHUTTLE DWG NO 4.7.2
KENTRON			DPS
LTR	PCN		44 X 34 PAGE 4-137 SHEET OF

2901.ART.7

5 MASS MEMORY UNIT

SECTION 5
MASS MEMORY UNIT

CONTENTS

Section		Page
5.1	OVERVIEW	5-1
5.1.1	MMU - Features/Performance	5-1
5.2	MMU HARDWARE	5-4
5.2.1	Tape	5-4
5.2.2	Tape Drive and Operation	5-4
5.2.3	Tape Layout	5-8
5.2.4	Heads	5-11
5.2.5	Error Detection and Correction	5-12a
5.2.5.1	ECC Capability	5-12a
5.2.5.2	ECC Hardware Implementation	5-12a
5.2.6	Interfaces - Electrical/Mechanical	5-12a
5.2.6.1	MMU Connector - J1 (Power)	5-12b
5.2.6.2	MMU Connector - J2	5-12c
5.2.6.3	MMU Connector - J3 (GSE)	5-15
5.3	OPERATION OF THE MMU	5-17
5.3.1	Initialization	5-17
5.3.2	Commands and Responses	5-17
5.3.3	Self-Fail Detection	5-20
5.3.4	BITE Status Words	5-21
5.4	INTERFACING SOFTWARE	5-25
5.4.1	Reconfiguration	5-25
5.4.1.1	Overlay MACRO	5-28
5.4.1.2	Incore Phase Table	5-30
5.4.1.3	Overlay Operations	5-30
5.4.2	Discrete Redundancy Management	5-31
5.4.3	Timeout Calculations	5-31
5.4.4	Command Processing	5-32
5.4.5	MMU Bus/Commander Selection	5-34
5.4.6	Completion Monitoring	5-35
5.4.7	BCE Programs	5-36
5.4.8	Error/Completion Processing	5-37
5.5	APPLICATIONS SOFTWARE	5-38
5.5.1	SM Common Buffer	5-39
5.5.2	MMU Read/Write	5-41
5.5.2.1	MMU R/W Instructions	5-42
5.5.2.2	Mass Memory Directory	5-50
5.5.2.3	Item Processor	5-50
5.5.2.4	Message Processor	5-53
5.5.2.4.1	MMU directory	5-53
5.5.2.4.2	Revision patch log	5-54
5.5.2.4.3	Read operations	5-54
5.5.2.4.4	Dump operations	5-57
5.5.2.4.5	Compare operations	5-57
5.5.2.4.6	Patch operations	5-58

Section Page

5.5.2.4.7 Write operations 5-60
5.5.2.4.8 Ready bits 5-61
5.5.2.4.9 Use of SM common buffer during R/W operations . . 5-62
5.5.2.4.10 Uplink procedure for MMU patch operations 5-62
5.5.3 OPS Transitions 5-62
5.5.4 SM Checkpoint 5-63
5.5.5 Crew Text 5-64
5.5.6 Roll-In Displays 5-64
5.6 MMU ERROR SIGNATURES 5-65
5.6.1 I/O Background 5-65
5.6.1.1 BCE 27 5-65
5.6.1.2 BCE 18 or 19 5-66
5.6.1.3 Ready Discretes and Timeouts 5-67
5.6.2 Data Output Errors 5-69
5.6.3 Access Time Error Signature 5-70
5.7 REFERENCES 5-71
5.8 MMU SOFTWARE FLOWS 5-71

MMU SOFTWARE FLOWS

Drawing Page

5.7.1 VMP_ITEM PROC 5-76
5.7.2 Message Processor Summary Flow 5-77
5.7.3 MM_MSG_PROC 5-79
5.7.4 MM_READ_LB 5-81
5.7.5 MM_LOAD_BLK_READ 5-83
5.7.6 MM_RPL_READ 5-85
5.7.7 MM_PATCH_MERGE 5-87
5.7.8 MM_DD_VALIDATION 5-89
5.7.9 MM_LOAD_BLOCK_W 5-91
5.7.10 MM_RPL_UPDATE 5-93
5.7.11 MM_LOADBLOCK_W 5-95
5.7.12 MM_LDB_PATCH_RESP 5-97
5.7.13 MM_BUFFER_COMPARE 5-99
5.7.14 MM_TO_MM_COMPARE 5-101
5.7.15 MM_PT_READ 5-103

SECTION 5
MASS MEMORY UNIT

5.1 OVERVIEW

The Space Shuttle Data Processing System (DPS) uses a data storage device called a Mass Memory Unit (MMU). This brief will discuss the mechanics of an MMU and how it responds to commands from the GPC's. It is not written with the intent of making the reader an expert on the details of every internal gate and register of the MMU, but rather to provide a working knowledge of the operation of the unit and its overall design. A functional schematic of the MMU is provided by the Space Shuttle Systems Handbook drawing of the MMU. The following areas will be addressed; the operating environment, tape, tape drive design and operation, tape layout, unit interfaces, initialization, commands, and self-fail detection.

There are three basic configurations of MMUs: the original orbiter MMU, the Spacelab MMU, and an upgraded orbiter MMU. The original orbiter MMU and Spacelab MMU are identical with the exception of power supplies. The Spacelab MMU power supply includes a power control relay that the orbiter MMUs do not have. The upgraded orbiter MMU was designed to improve the longevity of the MMU, decrease tape read errors, and cut down on the amount of magnetic tape failing qualification tests at the vendor during initial tape loading into the MMU. The upgraded MMUs include error detection and correction code, upgraded read heads, improved motor brushes, and reduced tape transport speed.

5.1.1 MMU - Features/Performance[1,3]

Manufacturer - Odetics Corporation/Anaheim, CA

Orbiter part no.: MC 615-0005

-201 Original MMU design with 3M900 series tape

-202 Original MMU design with 3M892 series tape

-300 Upgraded MMU with 3M892 series tape

Spacelab part no.: 4211700

Description - The MMU (fig. 5-2) is a coaxial reel-to-reel tape recorder capable of storing 800,000 hex halfwords (16 bits). (8,388,608 halfwords, 134 M bits.)

Orbiter location (fig. 5-1) - Mass Memory Unit #1 - AV Bay 1
Mass Memory Unit #2 - AV Bay 2

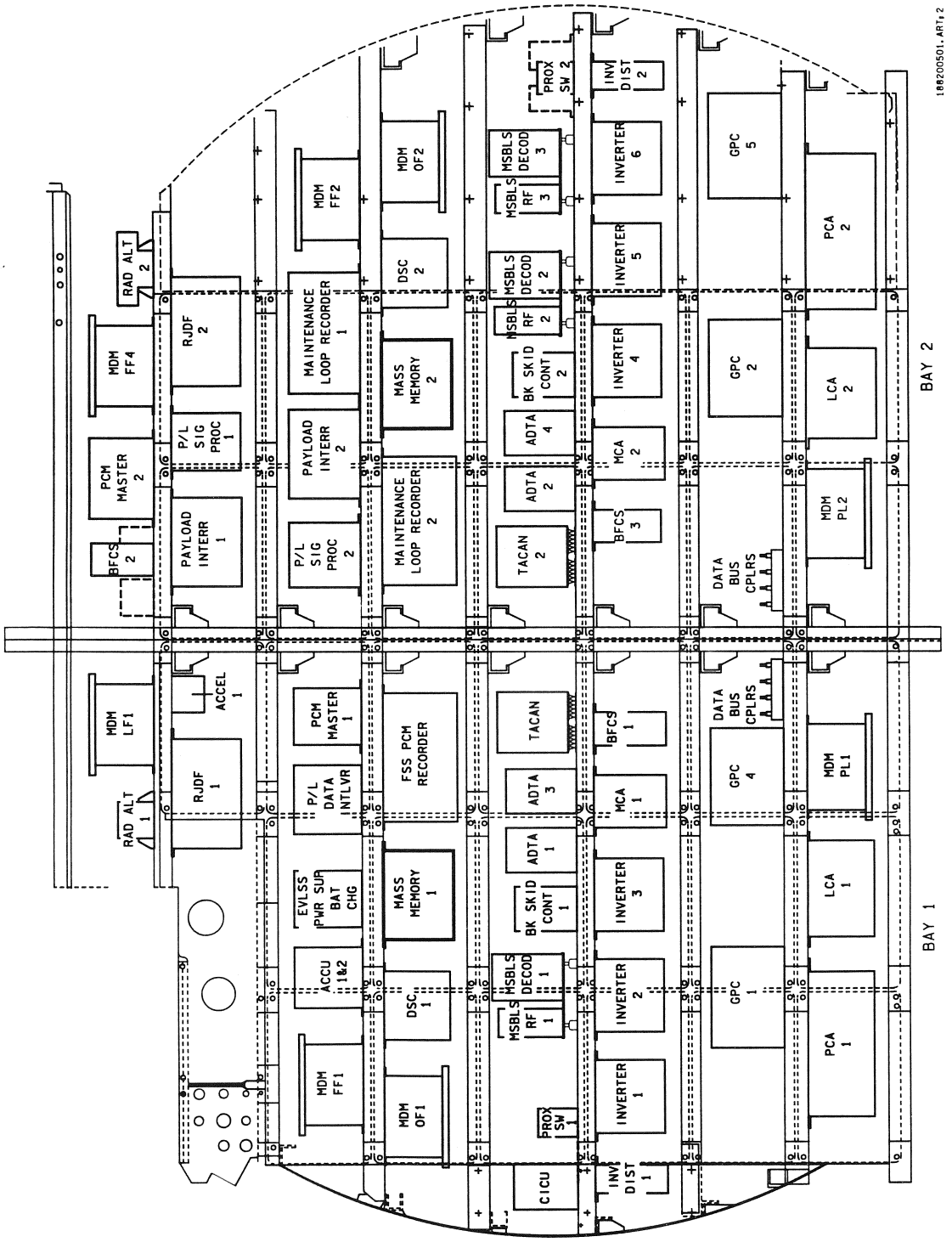


Figure 5-1.- Avionics bays 1 and 2.

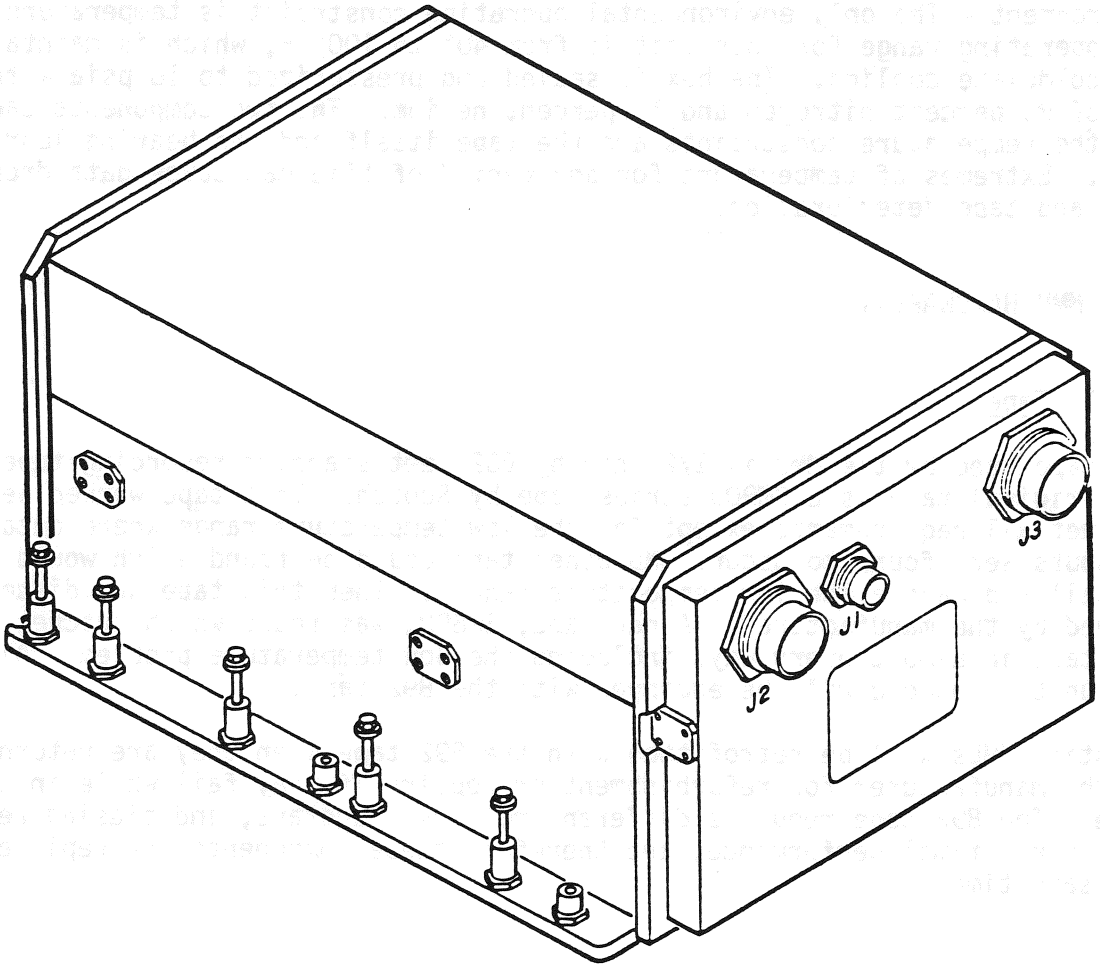


Figure 5-2.- The MMU.

Power - MMU	VMN	V CNTL
#1	MNA	CNTL AB1
#2	MNB	CNTL AB2

The MMU is powered with a 28 V dc feed of main bus power. (See section 5.2.4.)

Power consumption - When powered on, two states exist: standby and run.

MMU type	- 201, 202	- 300
Standby mode	16.8 watts	16.8 watts
Run mode	64.4 watts	72.3 watts

Size - Rectangular box 11.6 by 15.0 by 7.5 inches

Weight - 24.3/25.3 lb

Environment - The only environmental operating constraint is temperature. The operating range for this unit is from 40° to 100° F, which is maintained via coldplate cooling. The box is sealed and pressurized to 18 psia with a mix of 90 percent nitrogen and 10 percent helium. The two components causing the temperature constraints are the tape itself and the bearing lubricant. Extremes of temperature for any period of time can cause data dropouts and tape deterioration.

5.2 MMU HARDWARE[1,3]

5.2.1 Tape

The tape used by the MMU is 1/2 inch by 602 feet standard recording tape. The original tape is a 3M900 series tape by Scotch. This tape worked well and met all requirements except for the low temperature range where data dropouts were found to occur. No other tape could be found which would work as well and handle the low temperature ranges. Then this tape was discontinued by the manufacturer. A new tape, 3M892, was found which matches the 900 tape in almost every way, including the low temperature problem. All new units produced will be equipped with the 892 tape.

Existing MMUs will be retrofitted with the 892 tape when they are returned to the manufacturer for refurbishment or repairs if they fail while in service. The 892 tape requires different rollers, capstans, and biasing resistors for optimal performance, and therefore these components are replaced at the same time.

5.2.2 Tape Drive and Operation

The actual tape transport mechanism is rather unique. It is a reel-to-reel recorder, but the reels are located one on top of the other (coaxial) (fig. 5-3). Also, no system of reel takeup or rewind motors is used. The only tape drive is provided by three capstans. These capstans are driven by a dc motor through a series of three drive belts designed to provide zero side loading to the motor bearings and a triply redundant capstan drive (fig. 5-4). Tape tensioning and tapeup reel drive are provided through an assembly called a differential drive which is powered by a negator spring (fig. 5-5). The differential gear acts between the two reels. When tape is pulled from one reel by the capstans, the differential causes the other reel to turn in the opposite direction, thereby reeling in the tape. Tensioning the tape and accounting for differing effective reel diameters, as the tape moves, is done by applying a constant force to the differential using a large clock type spring called a negator spring. The most relaxed state of the spring is when the tape is equally divided between the two reels. As the tape moves either direction from this midpoint, the spring will be "wound up". Therefore, at the beginning or end of tape, the spring will be in its most tensioned state. The negator spring is a critical single point failure component. Its failure can be predicated quite closely only in terms of total number of cycles, but the layout of software on the MMU tape

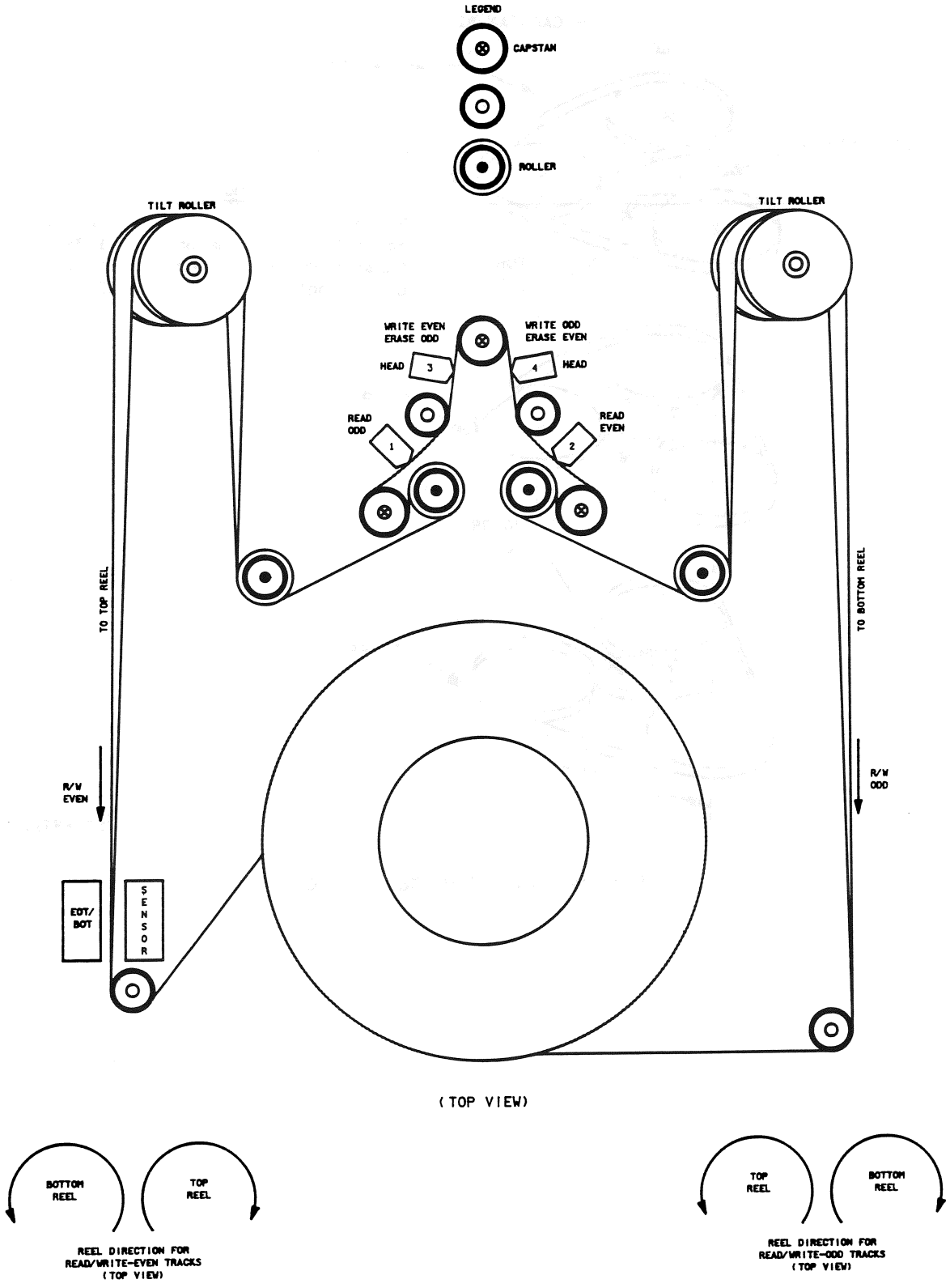


Figure 5-3.- Tape path.[3]

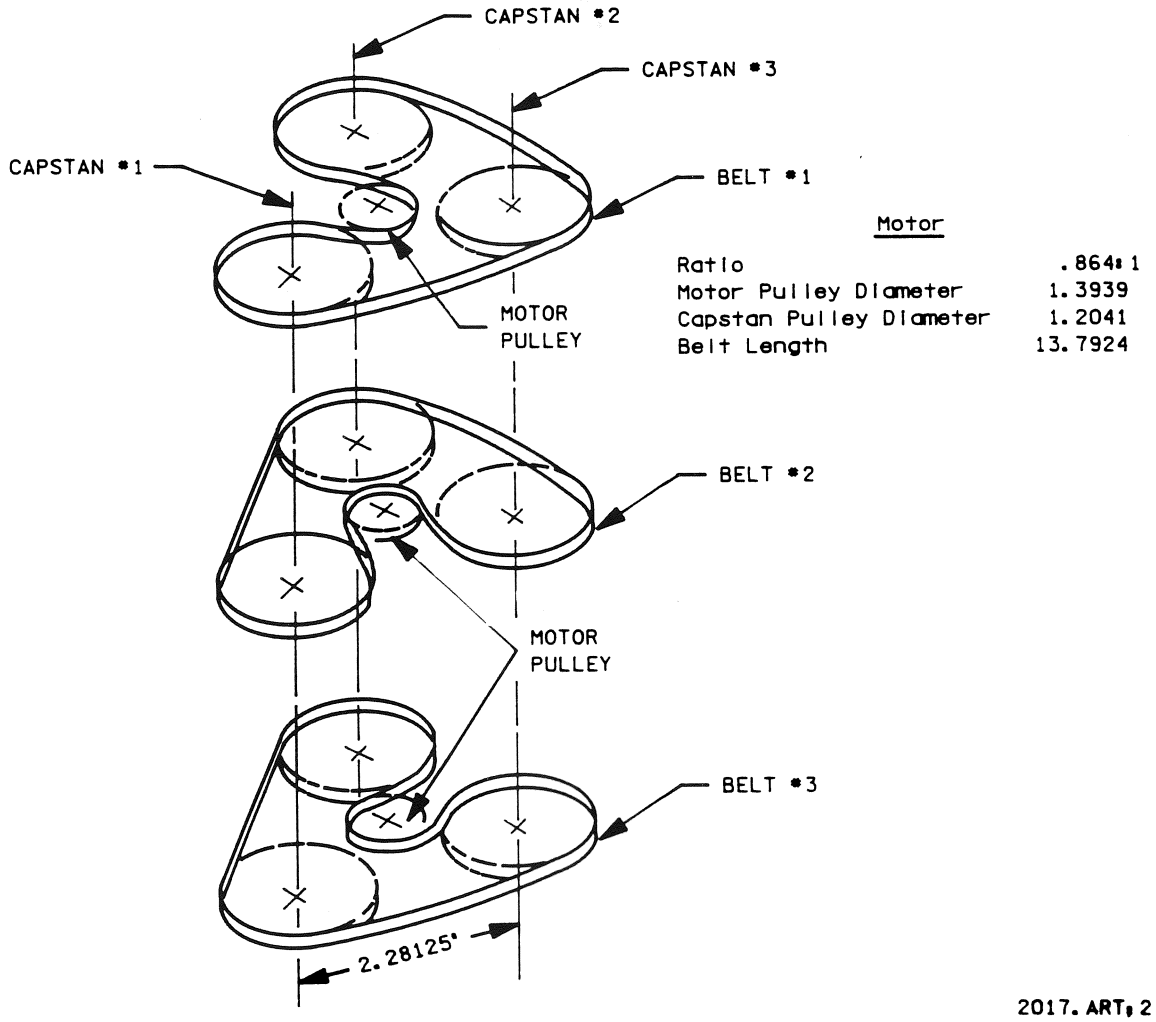


Figure 5-4.- Capstan drive.

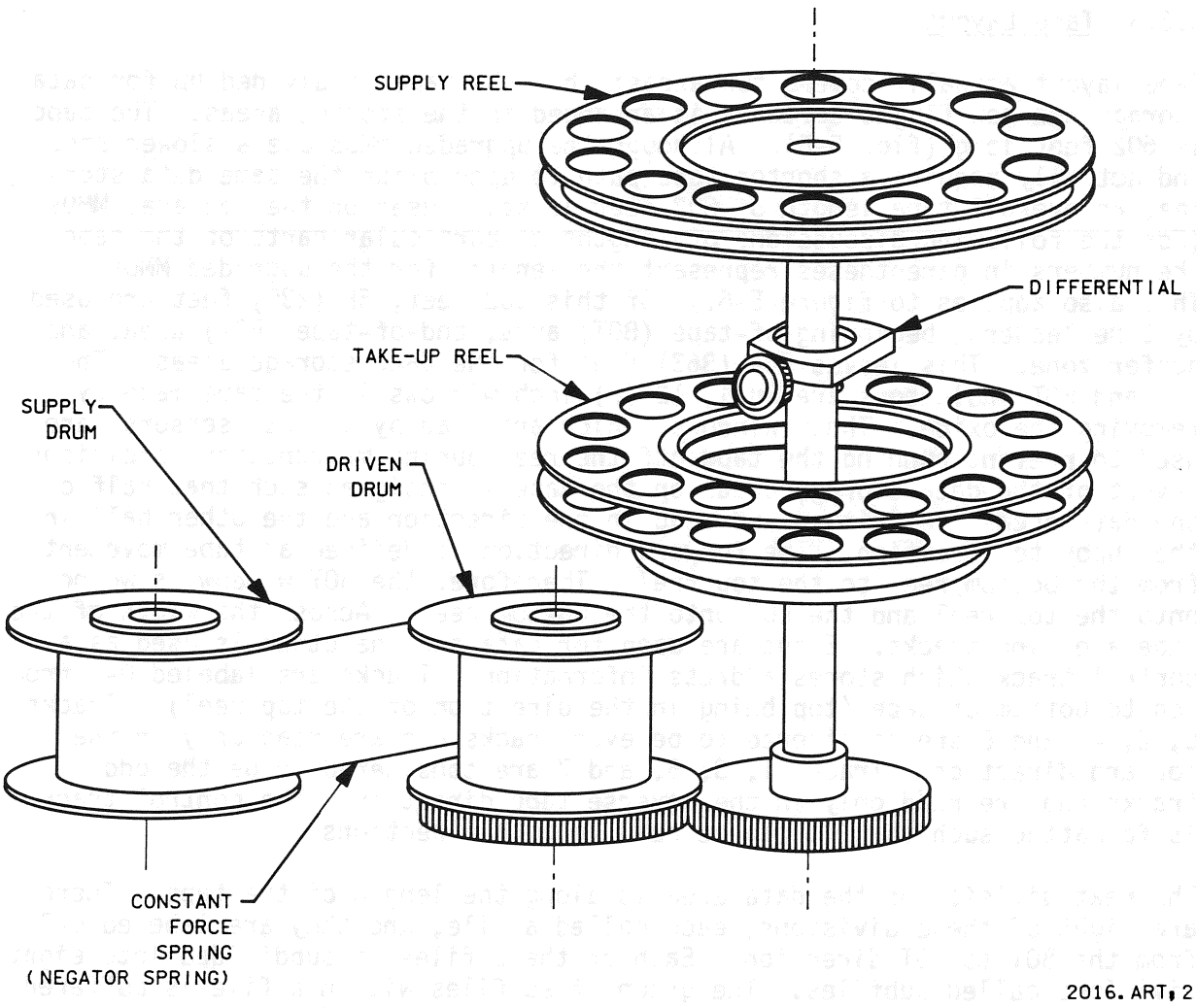


Figure 5-5.- Tape tensioning.

complicates this prediction. If the tape is repeatedly shuttled back and forth over a small section of the tape, this causes a cycling of the negator spring which counts against its expected life. Predicting or counting the total number of such cycles for an operational unit is next to impossible since no hardware counter exists to count the cycles. Therefore, the spring is due to be replaced at the same time the tape is replaced during the periodic maintenance. (This period has not yet been defined by JSC Engineering and Development (E&D)).

The tape speed in the original design MMU is 99.34 in/sec. For the upgraded MMU, the speed has been reduced by 30 percent to 66.23 in/sec. Because the internal clock rate used for reading and writing of data to the tape was not changed, the end result is an increased density of bits on the tape, but the actual data transfer rate onto or off of the tape remains the same. This works out to approximately 8.2 sec/file or about 0.82 sec/subfile when the gaps and end-of-file/beginning-of-file (EOF/BOF) areas are considered.

5.2.3 Tape Layout

Tape layout actually covers two areas: how the tape is divided up for data storage and how flight software is arranged in the storage areas. The tape is 602 feet long (fig. 5-6). Although the upgraded MMUs use a slower speed and actually require a shorter tape path to accomplish the same data storage, an overall tape length of 602 feet is still used on the upgraded MMUs. (For the following discussions of lengths of particular parts of the tape, the numbers in parentheses represent the lengths for the upgraded MMUs. This also applies to figure 5-6.) Of this 602 feet, 58 (239) feet are used by tape leaders, beginning-of-tape (BOT) area, end-of-tape (EOT) area, and buffer zone. This leaves 544 (363) feet for the data storage areas. The BOT and EOT indicators are dual, 12 (8)-inch windows in the tape made by removing the oxide. These windows, which are read by optical sensors, are used to prevent running the tape off the reel during malfunction conditions. Layout of the data storage areas on the tape is designed such that half of the data areas are written and read in one direction and the other half in the opposite direction. The forward direction is defined as tape movement from the bottom reel to the top reel. Therefore, the BOT window is wound onto the top reel and the EOT onto the bottom reel. Across the width of the tape are nine tracks. Eight are used for data and the other is used as a control track which stores address information. Tracks are labeled 0-7 from top to bottom of tape (top being in the direction of the top reel). Tracks 0, 2, 4, and 6 are considered to be even tracks and are read only in the forward direction. Tracks 1, 3, 5, and 7 are considered to be the odd tracks and are read only in the reverse tape direction. The control track is formatted such that it can be read in both directions.

The next division of the data area is along the length of the tape. There are eight of these divisions, each called a file, and they are labeled 0-7 from the BOT to EOT direction. Each of these files is subdivided into eight divisions called subfiles. The group of subfiles within a file is buffered on each end by an area called either a BOF or an EOF. These areas are the same size as the actual subfile. The order of subfile numbering is determined by the track numbers and hence the direction of travel. Even track subfiles are labeled BOF 0, 1, 2, 3, 4, 5, 6, 7, EOF from the BOT direction to EOT. Odd track subfiles are labeled in the reverse order EOF, 7, 6, 5, 4, 3, 2, 1, 0, BOF.

Each subfile is further subdivided into 32 areas called blocks (fig. 5-7). Again, these are labeled in the direction of travel. Each block of data holds 512 data words. For the original MMU these are 17-bit words (16 data bits and 1 bit for odd parity). For the upgraded MMU there are four groupings of 128 words and 128 bits for a checkword. There is a buffer area between blocks of data. This area contains unique data words written at the beginning and end of the gap that are used by the read logic in determining where the block data ends and the gap area begins. Also located between these special words is an alternating pattern of ones and zeros written at half the normal rate to identify the gap area. The buffer area takes up one-third of the available space. Data words are written onto the tape in a format called biphase low (BI-Ø-L). This type of format always has a transition in the center of each bit cell. A zero is represented by a low-high

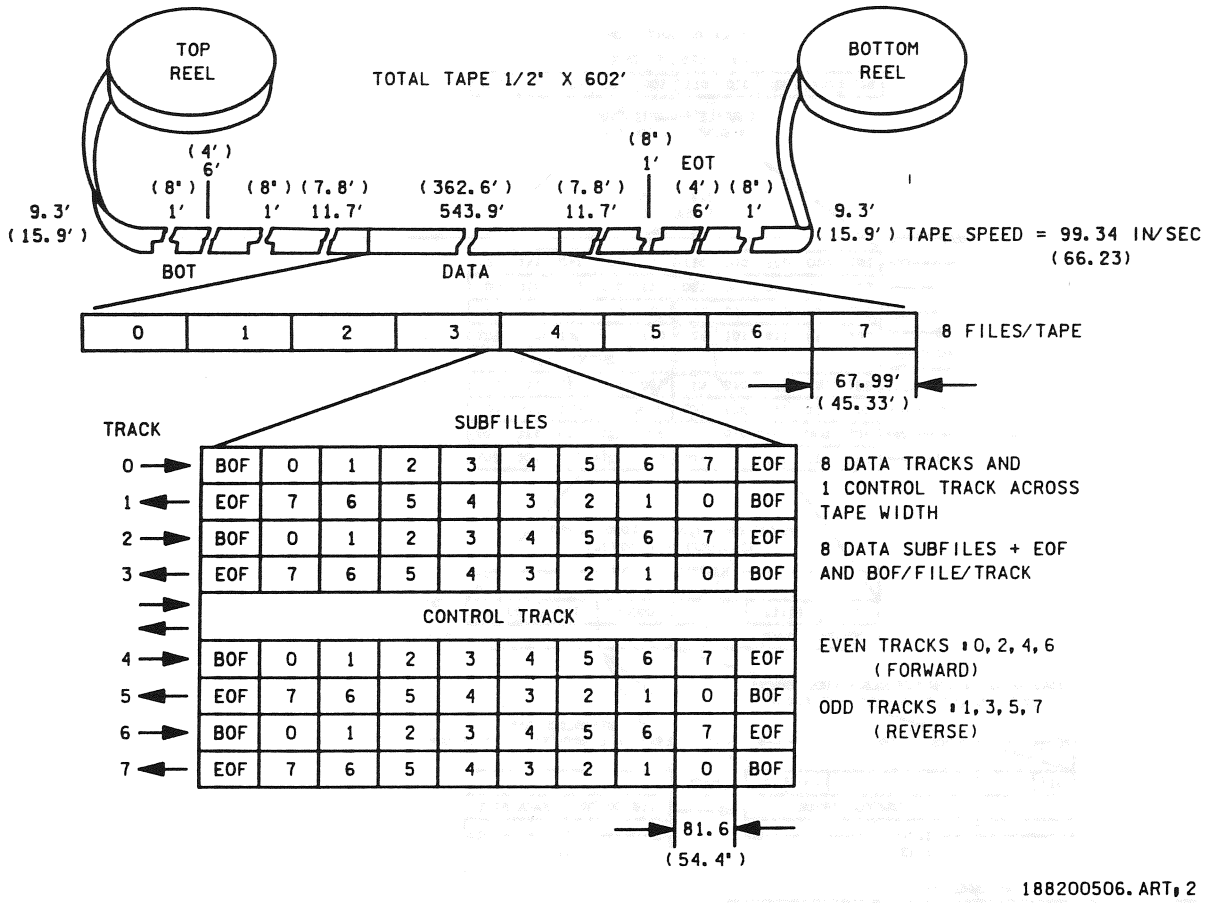


Figure 5-6.- Tape organization.[3]

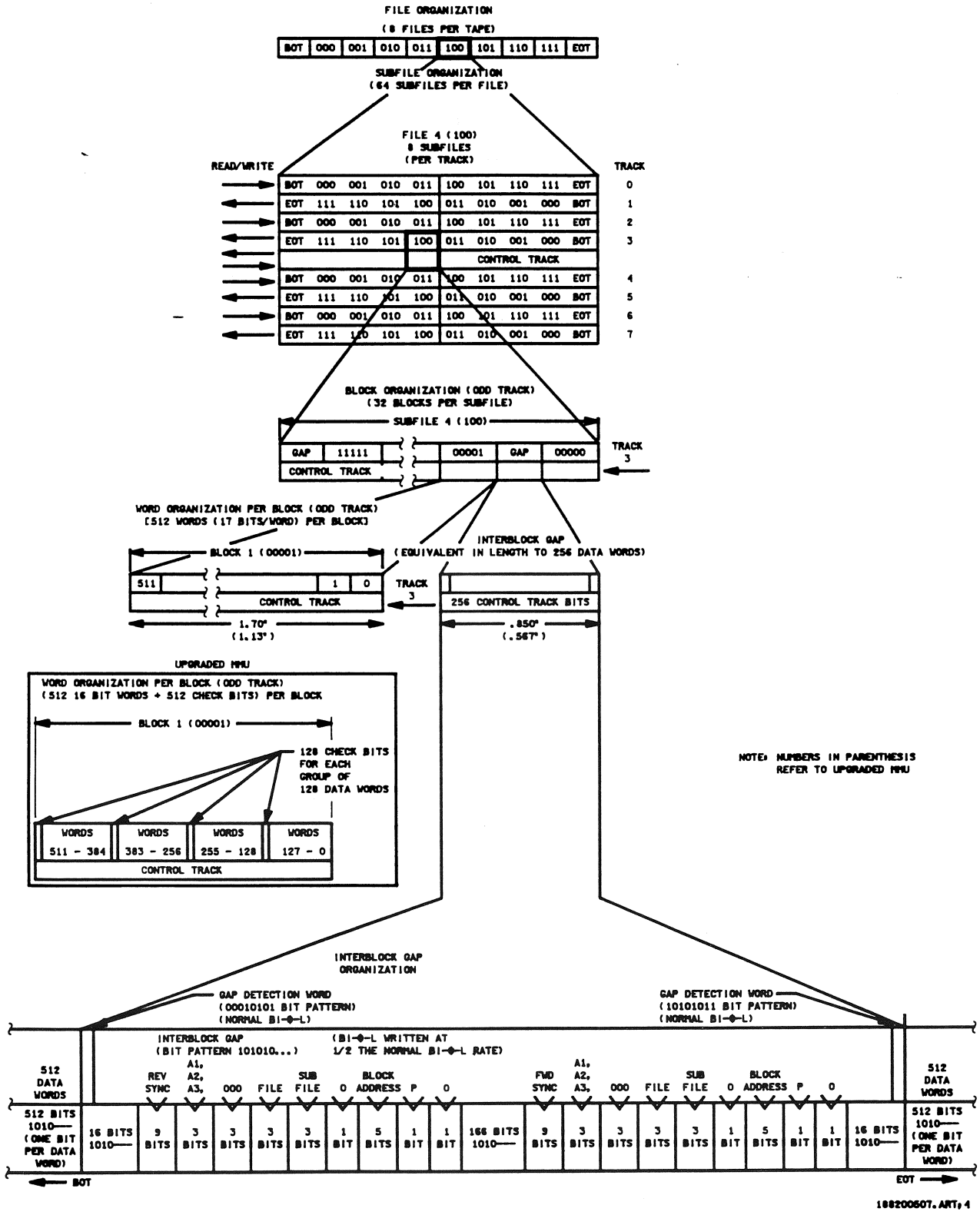


Figure 5-7.- Track layout.3

pattern and a one by a high-low pattern. Always having a transition in the center of the bit cell makes this a self-clocking type of format (refer to fig. 5-10).

Across from this buffer area (or gap) is where the address information is located on the control track. This control track area contains addresses for the preceding and following data blocks. Each address contains a forward or reverse sync code, file number, subfile number, and block number. The subfile information may be replaced by an EOF or BOF code when appropriate. The forward and reverse sync codes allow the control track decoding logic to read the control track in either direction and utilize the appropriate address. Stop bits may also be included in the address when appropriate. They provide a cue for the tape to stop at the desired place within a given subfile (fig. 5-7).

In summary, the tape layout provides 8 tracks of 8 files of 8 subfiles, of 32 blocks, each containing 512 words for a total storage capacity of 8,388,608 16-bit data words. The data is addressable by blocks of 512 words.

5.2.4 Heads

Two types of heads are used on the MMU: read heads and write/erase heads. Head no. 1 reads the control track and odd tracks only. Head no. 2 reads the control track and even tracks only. Head no. 3 writes even tracks and erases for odd tracks. Head no. 4 writes to odd tracks and erases for even tracks (fig. 5-3). Only one of the data tracks per head is energized at one time.

The wear experienced by the read and the write/erase heads has been different. Because the write/erase heads write one set of tracks and erase the other set, the head must have an active segment for all eight data tracks and the control track. Conversely, a read head is only used to read the even or odd tracks and therefore has only four active segments in addition to the control track segment (fig. 5-7a). The areas on the read head between the active segments are twice the size of those on the write/erase heads. Because of the abrasive nature of the 3M392 tape, the soft core material of the heads tends to wear, diminishing the useful life of the read heads to 1/2 to 1/3 that of the write/erase heads. When MMUs go in for repair, the read heads are often worn down to <50 percent life and are replaced. The write/erase heads have never needed replacing.

To remedy the head wear problem, the upgraded MMU includes new extended life read heads. These heads have additional alfesil strips inserted between the active segment on the head surface, providing a read head life similar to that of the write/erase heads.

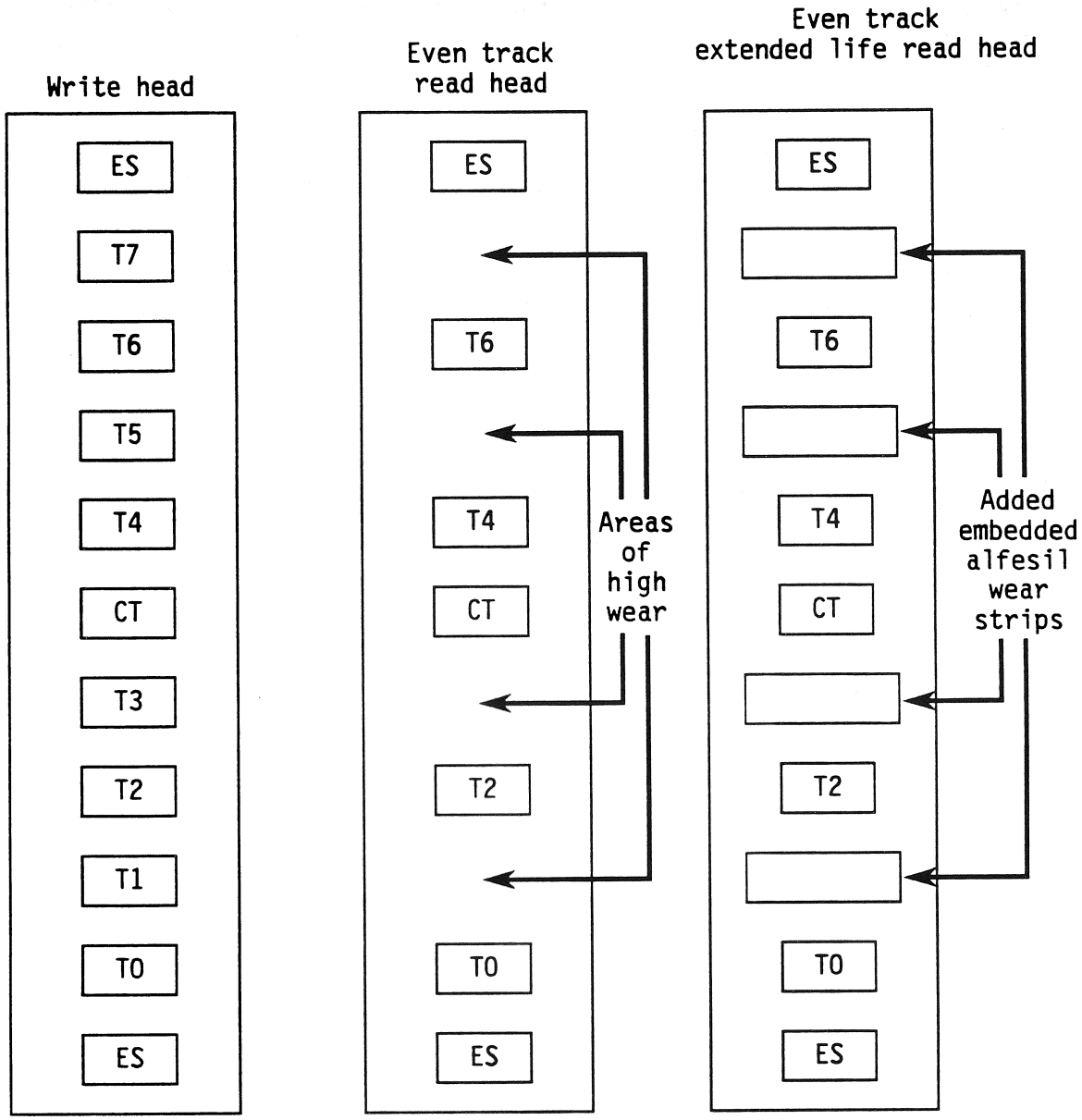


Figure 5-7a.- MMU heads.

5.2.5 Error Detection and Correction

The upgraded MMUs (-300 series) have a new circuit assembly that interfaces with the existing circuitry to provide a method of detecting and correcting errors in data read from the MMU tape. This capability was added to extend MMU use life, eliminate nonrepeatable intermittent errors occurring in the field, and minimize the rejection of tape for fixed location tape blemishes during factory loading of tape onto the MMU. This error correction capability will be labeled as the MMU Error Correcting Code (ECC) assembly or ECC module.

5.2.5.1 ECC Capability

The ECC is designed to be able to detect and correct an error burst of up to 48 bits in length within a group of 128 data words with 0 percent error in the correction. (Note that this is equivalent to a scratch that is 0.0094 inch wide in a tape area that is 0.42 inch long.) It can handle bursts of up to 80 bits, but the probability of having an error in the corrected bits goes up to 10^{-11}. For a burst of 110 bits, the probability of error goes up to 10^{-2}. For any error burst of >48 bits, bit 4 of the MMU status word B is set to indicate the error string. Since error strings of 48 bits or less are always corrected, there is no external indication of the encounter.

5.2.5.2 ECC Hardware Implementation

The ECC is a hardware implementation rather than a software implementation. A new ECC circuit assembly was added to the MMU in a previously vacant location and changes were made to the existing read/write electronics to interface the ECC with the read and write data streams. The new assembly contains two circuit boards, adds 7.9 watts to the MMU power consumption, and adds 1 pound to the MMU weight.

The original MMU data organization on the tape used a parity bit with each 16-bit data word for a total of 512 17-bit words per MMU block. The ECC implementation splits each MMU block into four groupings, each consisting of 128 16-bit data words followed by a 128-bit checkword. The ECC generates this 128-bit checkword when writing data to the MMU and uses the checkword when reading data from the MMU to find errors and correct them.

5.2.6 Interfaces - Electrical/Mechanical

The MMU interfaces electrically with the vehicle through connectors J1, J2, and J3 and mechanically through the coldplate mount for cooling. MMU's 1 and 2 are located in AV bays 1 and 2, respectively. Each unit has 10 captive bolts to fasten it to the coldplate.

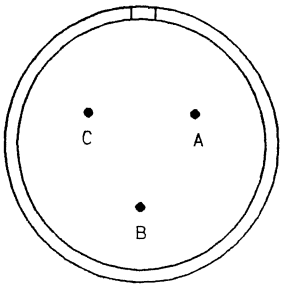
I 5.2.6.1 MMU Connector - J1 (Power)

The J1 connector is the power connector. It contains the 28 V dc input and return lines. The power is supplied by main bus A or B (MMU 1 or 2, respectively) through an RPC assembly. The control switches are single-pole, single-throw lever switches and are located on cockpit overhead panels O14 and O15. The control power is from a CNTLAB 1 and CNTLAB 2. Notice that each MMU is uniquely powered by a main and control bus with no redundancy. Telemetry is on the control leg of the RPC, not the output, and therefore does not indicate that power is actually applied to the MMU (i.e., if the main bus or RPC is out, the telemetry indicates that the MMU switch is on, but the MMU will not be powered).

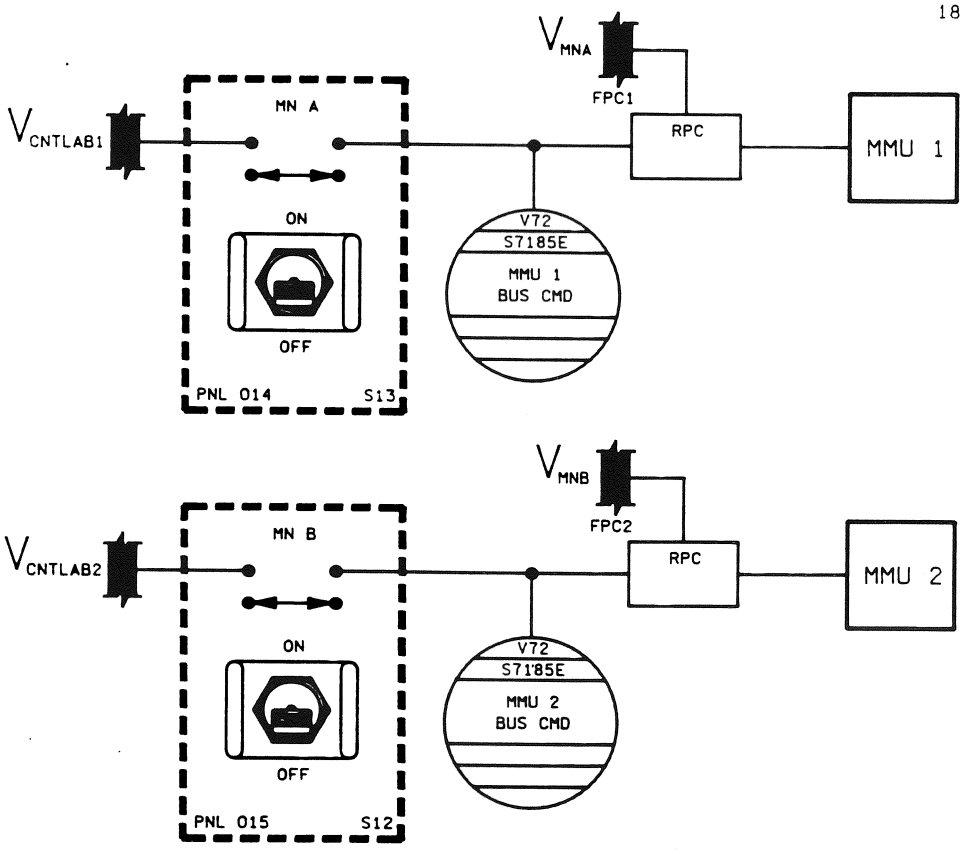
J1: Power connector[3]

M83723-12H1203N
Code indent: 11139
Mating connector
P/N AFD56-12-03-SN-1A

Pin	Function
A	+28 volt dc, Input
B	+28 volt dc Return
C	Chassis ground



188200509.ART, 1



2012.ART, 2

Figure 5-8.- Orbiter/MMU power.

5.2.6.2 MMU Connector - J2[3]

Connector J2 carries the MMU discrete signals and monitoring lines as well as the data bus to the GPC's.

J2: I/O connector

M83723-12H2255W
Code ident: 11139
Mating connector
P/N AFD56-22-55-SW-1A

Pin	Function	Pin	Function
A	Ready 1 (True)	f	Read - 12 V monitor
B	Ready 1 (Compl)	g	Standby - 12 V monitor
C	Ready 2 (True)	h	Signal common
D	Ready 2 (Compl)	i	Reset no. 5 (Compl)
E	Ready 3 (True)	j	Reset no. 4 (True)
F	Ready 3 (Compl)	k	Reset no. 2 (Compl)
G	Spare	m	Reset no. 2 (True)
H	Temperature	n	Reset no. 1 (Compl)
J	Pressure	p	Spare
K	Analog trend	q	Spare
L	Standby +5 V monitor	r	Spare
M	Standby +12 V monitor	s	Spare
N	Reset no. 5 (True)	t	Spare
P	Reset no. 3 (Compl)	u	Servo -20 V monitor
R	Reset no. 3 (True)	v	Servo +20 V monitor
S	MIA Bus (True)	w	Read +5 V monitor
T	MIA Bus (Compl)	x	Spare
U	Reset no. 1 (True)	y	Spare
V	Spare	z	Reset no. 4 (Compl)
W	Ready 4 (True)	AA	1 MHz data shield
X	Ready 4 (Compl)	BB	Spare
Y	Ready 5 (True)	CC	Spare
Z	Ready 5 (Compl)	DD	Spare
a	Spare	EE	Read +12 V monitor
b	Spare	FF	Spare
c	Spare	GG	Spare
d	Spare	HH	Spare

The primary discrete is the READY discrete which indicates whether the MMU is powered up and/or not busy processing previous commands. In particular, the MMU must not be in an initialization mode (post powerup), be transmitting data, be moving the tape, or have a command present. There are five READY discrete lines per MMU, one line for each GPC. The lines are the standard discrete true/complement pair type used throughout the DPS.

This page intentionally left blank

Another set of five discretes is the RESET discretes. These are not connected for Orbiter use but are connected for Spacelab use. The RESET discrete can cause a resetting of the MMU electronics similar to that done on initial powerup, except that no tape positioning will occur.

The data bus on J2 is a standard twisted pair 1 MHz data bus which connects the MMU to all five GPC's. There is a separate data bus that connects the other MMU to the GPC's through different IOP BCE's. The bus used for MMU 1 is labeled 18 and the bus for MMU 2 is labeled 19 (fig. 5-9). The data bus is used to send commands requesting certain responses to the MMU. Data being read from, or to, the MMU travels over this data bus.

The remaining functions located in the J2 connector are MMU monitoring lines. None of these are connected or used by the Orbiter. Spacelab does use the temperature transducer, the analog trend from the read amplifier, and the +5 V STBY power supply monitor line (pins H, K, and L).

In addition, for Spacelab MMU's, two other sets of signals are carried by the J2 connectors. They are control and monitor lines for a power control relay that only the Spacelab MMU power supply contains (fig. 5-10). The relay is a latching type relay with an additional set of contacts to monitor the relay state. The relay control commands come from Remote Acquisition Unit (RAU) B (an RAU is like an MDM) and Payload Forward 1 MDM (PF1) through the Remote Acquisition and Advisory Box (RAAB). The relay status is picked up by both PF1 and RAU B through the RAAB. This relay is used to power up the Spacelab MMU instead of a panel switch. The subbus that the MMU is powered from, CDMS 4-1, is not individually switchable.

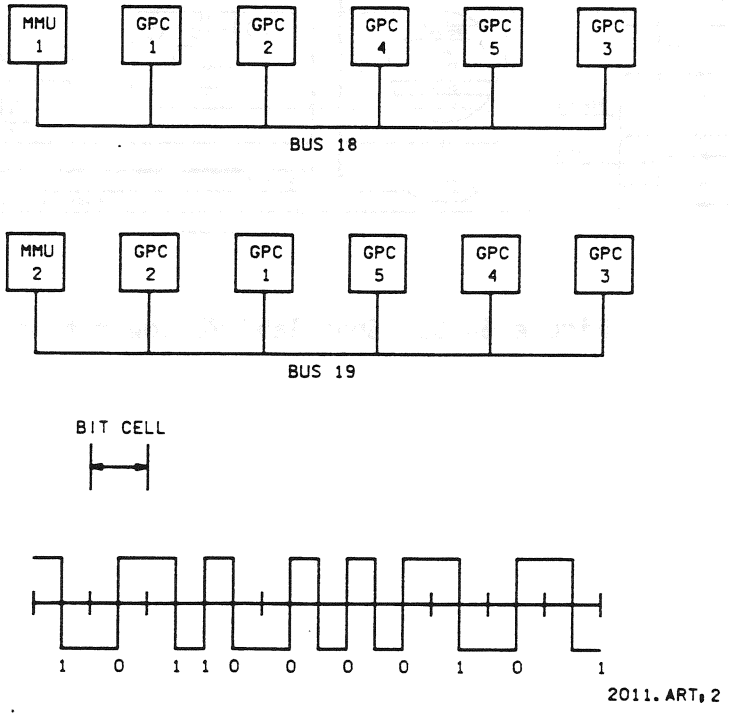


Figure 5-9.- Data bus and biphase-L code.

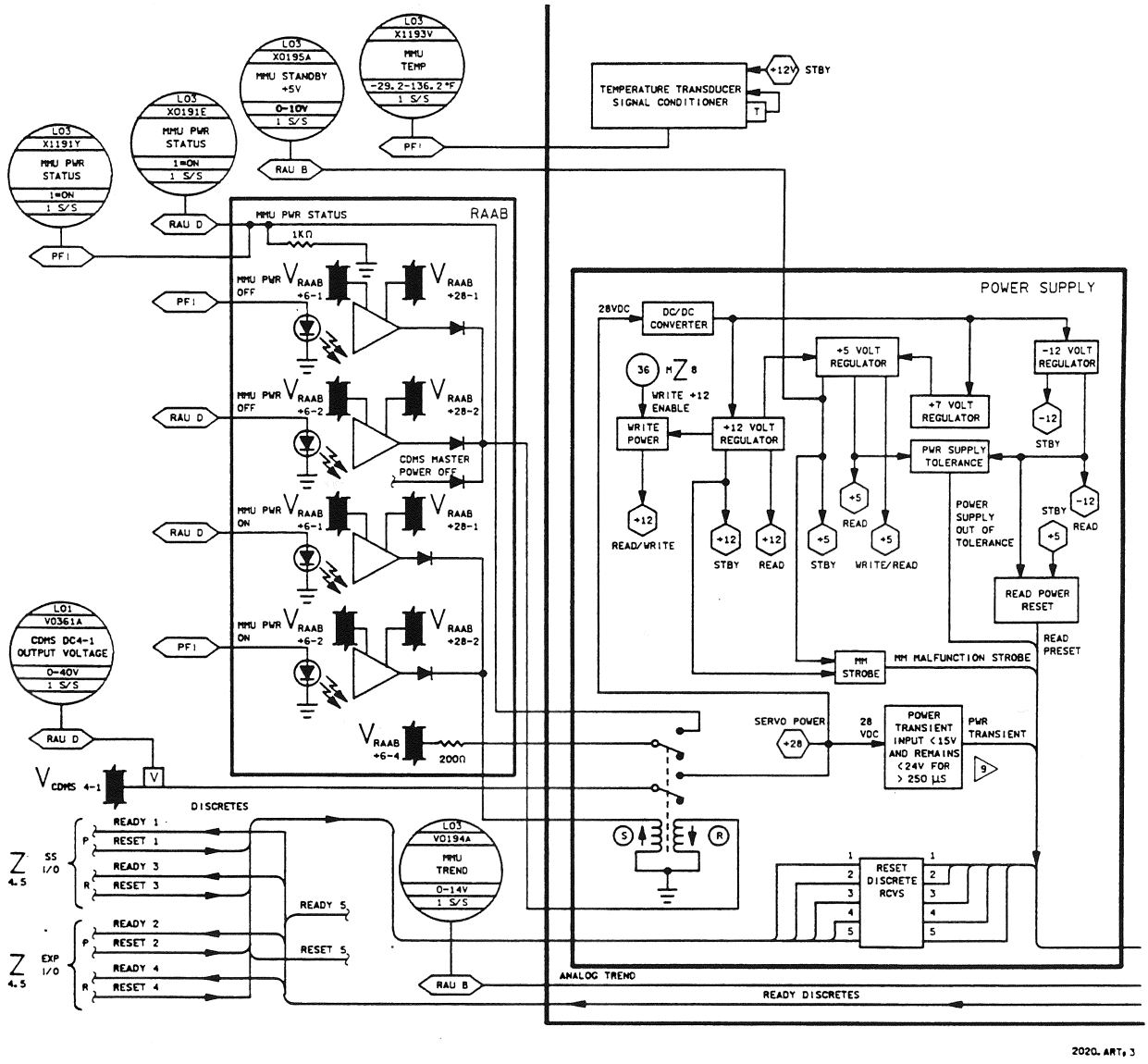


Figure 5-10.- Spacelab/MMU power feed.

5.2.6.3 MMU Connector - J3 (GSE)

The third connector, J3, is mainly a ground checkout/use connector except for two functions. Five pins are used to encode an MIA identification address. Pins connected to ground are a logic "0" and unconnected pins are a logic "1". The MIA code for the shuttle MMU is "11" (01011) and the Spacelab MMU code is "21" (10101). All commands to the MMU carry this code. Another group of eight pins provides for enabling the write function for each data track by grounding the corresponding pin. For Orbiter and Spacelab, all tracks are enabled. The rest of the pins on the J3 connector are for ground testing of the MMU with a special tester. These test points allow the internal logic to be sampled at selected points and also provide for writing of the control track on new tape. Since the only pins used for flight on the J3 connector are the grounds for write track enabling and MIA code establishment, a connector plug is used which as the appropriate connecting wires from pins to ground.

J3: GSE connector

M83723-12H2255N
Code ident: 11139
Mating connector
P/N AFD56-22-55-SW-1A

Pin	Function	Pin	Function
A	Test point no. 1 output +	f	Test point select input B
B	Test point no. 1 output -	g	MIA identification 0
		h	MIA identification 1
C	Test point no. 2 output +	i	MIA identification 2
		j	MIA identification 3
D	Write control track +	k	MIA identification 4
E	Erase control track +	m	Signal common
F	Write control track enable	n	Test point select A
		p	Initialize indication
G	Signal common	q	Erase control track -
H	Analog data track	r	Write control track center tap
J	Signal common		
K	Analog control track data	s	Erase control track center tap
L	Write track no. 0	t	Spare
M	Write track no. 1	u	Write BITE
N	Write track no. 2	v	Spare
P	Write track no. 3	w	Erase BITE
R	Write track no. 4	x	Test point select input C
S	Write track no. 5		
T	Write track no. 6	y	Test point select input F
U	Write track no. 7		
V	Test point no. 2 output -	z	Test point select input G
W	Forced tach lock	AA	Spare
X	Spare	BB	Position tape ind.
Y	Spare	CC	Spare
Z	Write control trk -	DD	Servo error
a	Spare	EE	Main power supply overtemp
b	Test point pwr enab		
c	Spare	FF	Test point select input D
d	Spare		
e	Test point sel +5 V	GG	Test point select input E
		HH	Servo overtemp

5.3 OPERATION OF THE MMU

5.3.1 Initialization

Initialization occurs every time the MMU is powered on or if a power transient of >250 microseconds occurs. When this occurs, the logic registers are reset and the tape is driven in a forward direction in order to read the location from the control track. If the center of the tape (BOF 4) is ahead of the present location, the tape will continue until it reaches the center (up to 35 sec). If the center is in the reverse direction from the present location, the tape will stop, reverse direction, and drive until the BOF 4 is reached. At this time, the unit goes into a standby mode, which means that the only internal power circuits energized are those required to receive and recognize a command. Whenever a command is received, the main power supply is turned on and the unit services the request. Whenever the MMU is not busy responding to a command, it will be in the standby mode.

5.3.2 Commands and Responses

As explained in the interface section, each MMU has a dedicated data bus to the GPC's for the transfer of data. This twisted pair bus carries the commands and responses at a rate of 1 MHz. Each command word (traveling to MMU) and response word (traveling from MMU) is composed of 28 bits of data (fig. 5-11). The first three bits of the word are taken up by either the data or command sync pattern. This is a special pulse pattern used by the Multiplexer Interface Adapters (MIA's) located in the MMU and GPC IOP to recognize the start of a command or data word. (An MIA interfaces between the data bus and the hardware on each end of the bus.) Bits 4-8 are used to contain the device ID information which is a decimal 11 for an Orbiter MMU and a decimal 21 for the Spacelab MMU. Bits 9-27 vary depending on the specific command or response. Bit 28 is used for odd parity.

With the exception of the power on/initialization sequence, the MMU responds only to commands it receives. There are seven different commands that the MMU responds to. They are read, write enable, extended data, position tape command, position tape request, and status request (fig. 5-12). These are the individual commands that BCE programs build to control the MMU. These individual commands are grouped together to perform specific transactions. The Orbiter GPC's recognize four types of MMU transactions; prepositioning, reading, reading in an overlay, and writing to MMU. Each one of the individual commands will be discussed. The transaction concept will be discussed in the software section of this brief.

The position tape command causes the tape to move to a specific subfile (or BOF or EOF area) of a particular file. It also carries the code for the track to be read and stores this in a register. Due to the time involved for the tape to come up to speed, all position commands move the tape to the middle of the subfile preceding the subfile to be read from or written to. When the read command is received, the tape starts moving and is up to speed by the time it reaches the subfile where the reading is to be done. The

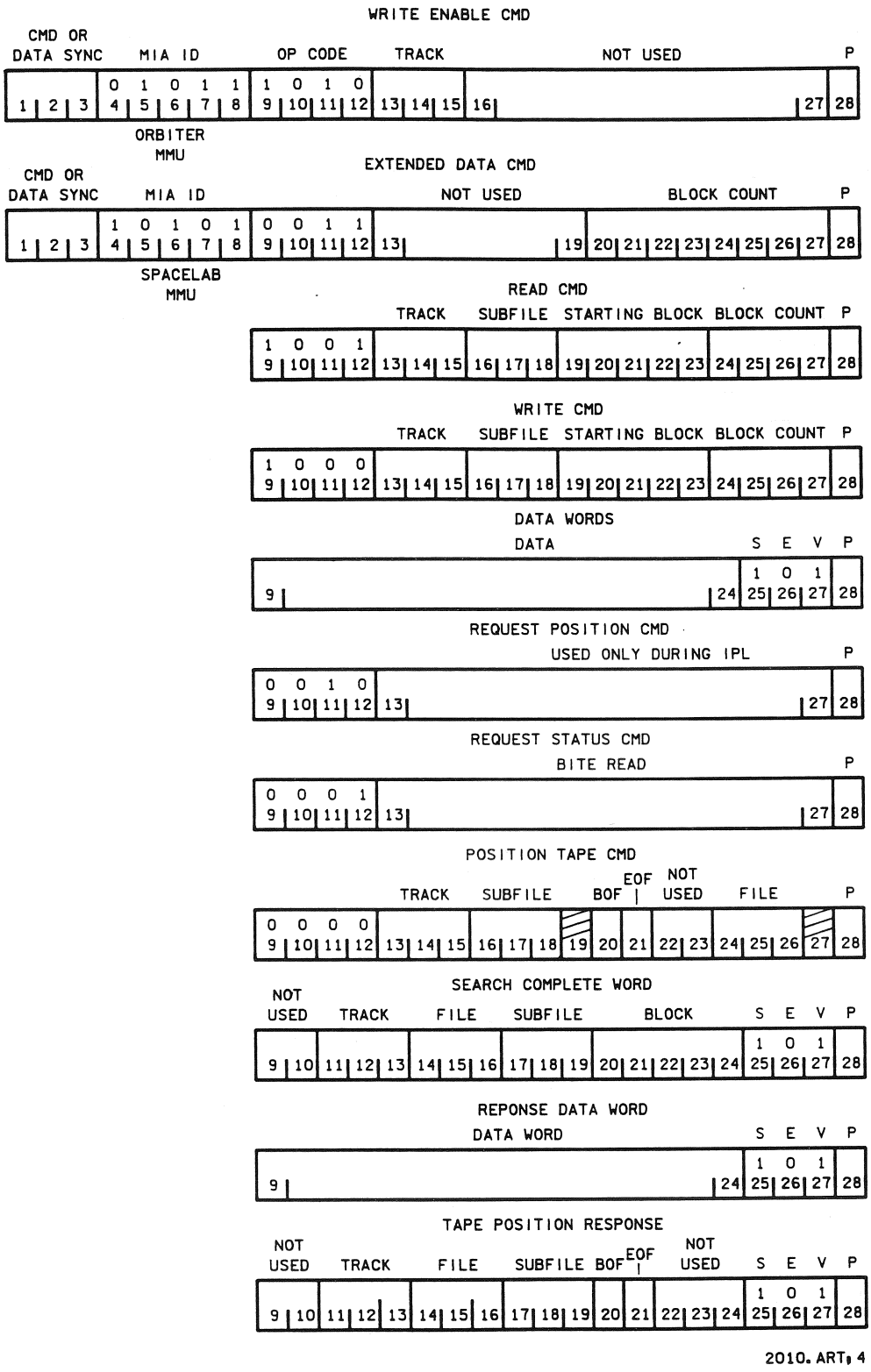


Figure 5-11.- MMU commands and responses.

subfile calculation is done in a FCOS MMU program called FIOMGDSP (MM GTG-I/O-Dispatcher). Similarly, when stopping, the tape is driven/coasts to the middle of the subfile following the last subfile read from or written to.

The read command requests that the MMU read a specified number of blocks (from 1-16), starting with a specific block, from a specified subfile and track. The file to be read from is the file that the position command (which precedes a read command) specified. This command causes the tape to start moving and comparing addresses. When the required address is found, the reading starts and data words are transmitted over the data bus. Because of the interblock gap between blocks of stored data, there is an 8.58 ms gap on the data bus between every 512 words. More than 16 blocks of data can be read if an extended block count command precedes the read command. Using this, up to 256 blocks (1 file) may be read with one read command.

The extended block count command is used in conjunction with the read and write commands in order to enable more than 16 blocks of data to be read from or written to the MMU in one operation. The command specifies the number of blocks to be read or written, up to 256 blocks (up to 1 file).

Preceding write operations is a command called write enable command. This must be used to enable the write operation for a specific track.

The write command itself is similar to the read command except for the OP code. It specifies the number of blocks, the starting block, subfile, and track. These areas must be in the file specified by the preceding position command. Like the read command, if an extended block count is to be used, the block count area of the write command is not used.

Because the MMU takes time to get the tape up to speed and reach the beginning of the block to be written to, the GPC needs to know when that point is reached in order to start transmitting data to the MMU. The MMU uses a word called a Search Complete Word (SCW) to signal the GPC. This word is transmitted to the GPC 32 word times before the start of the write area is reached. When the GPC receives the SCW, it starts sending data to the MMU.

A no-movement command is a tape position request command. This command causes the MMU to ship out a word containing the track, file, and subfile (or BOF or EOF) address of the current tape location. It reads this out of a control track location storage register and therefore requires no tape movement. The PASS software does not use this command. It is used during the IPL sequence and the BFS loading sequence to verify successful completion of the load.

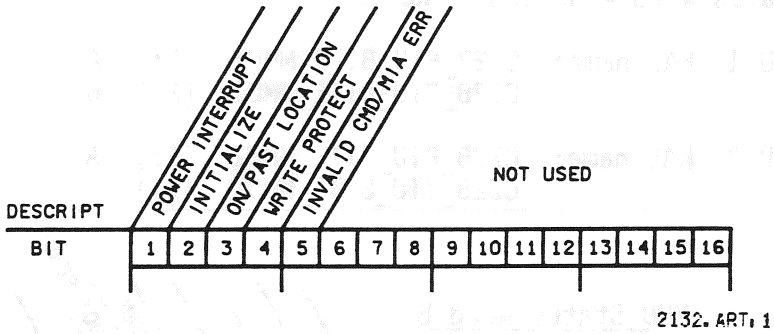
A final type of command is the status request command. This causes the MMU to ship out the contents of the two fault detection registers, A and B, and causes the registers to be reset. The SEV bits in the transmitted word containing the status register contents is forced to the 101 pattern for this case only to ensure receipt by the GPC. Status register A is transmitted first, followed by register B.

5.3.3 Self-Fail Detection

The MMU is equipped with internal logic to detect error conditions and store that information in the two status registers A and B. The BITE is designed to detect internal malfunctions as well as external problems such as to improper commands. Register A only uses 5 bits to describe external problems. Register B has 16 bits for the internal malfunctions. (See section 5.3.4 for a listing of the individual bits and their meanings.) Whenever a bit is set in one of the registers, the validity bit (V) in the following response messages is set to zero to indicate that an error has occurred. Status words are read by GPC's at the beginning and ending of each MMU transaction. Each time the registers are read, they are reset. The SEV bits previously mentioned are part of a type of fault detection system used on the shuttle by MMU's and all other Bus Terminal Units (BTU's). The SEV bits are 3 bits in the command words from GPC's to BTU's and in response words from BTU's to GPC's. The normal pattern for bits is 101. The S bit is to indicate power transients or initial MMU powerup (preset conditions). The E bit is for Multiplexer/Demultiplexers (MDM's), and the MMU hard-wires this to zero. The V bit is a validity bit and is set to zero if internal MMU problems have been detected which would also have set another status register bit.

5.3.4 BITE Status Words[3]

MMU Status WD A



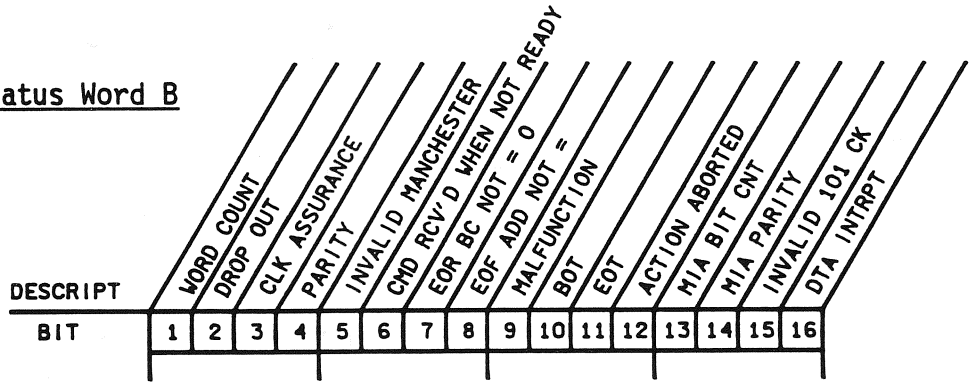
Bit		Description
1	Power On Preset/Power	Occurs for power interrupts of 250 μs. The S-bit is set to zero and the V-bit is set to one. Upon power application, initializes to BOF.
2	R/W Address ≠	Commanded file address ≠ present position file address. This causes a reinitialization and the V-bit to be set to zero. A BCE timeout will occur.
3	On/Past Location	The commanded address is within or behind the current subfile address. A BCE timeout will occur.
4	Write Protect	This occurs when a write command is issued if: • The track selected is protected against writing (this is not being done at present). or • The track requested to be written to is not the track selected and stored in MM by a previous write enable command or • Bit 2 of status word A is set. or • Bit 1 of status word A is set. A BCE timeout will occur.
5	Invalid CMD/MIA Error	Set if an invalid OP code is received, a position tape with both BOF and EOF bits set, MIA parity error, MIA invalid Manchester, or a MIA bit count error occurs. A V-bit is set to zero.
6-16		Not used.

Status Word - Software Names

MMU 1 HAL name: CZ2B_BTU_BITE_MMU1 (1:) A
CZ2B_BTU_BITE_MMU1 (2:) B

MMU 2 HAL name: CZ2B_BTU_BITE_MMU2 (1:) A
CZ2B_BTU_BITE_MMU2 (2:) B

MMU Status Word B



2132.ART, 1

Bit	Description	
1	Write Word Count Error	Set if <512 words/block are received for a write operation. Unit continues writing until the end of the block and then stops in the next subfile. The V-bit is set to zero.
2	Dropout	Indicates a read tape data dropout. The read will continue but the V-bit is set to zero.
2 (upgraded MMU)	Dropout	Detects read tape dropout for GSE test function only. Active only if error detection and correction code is disabled.
3	Clock Assurance	Set if the total number of clock pulses per block of data read ≠ (512 x 17). The read continues, but the V-bit is set to zero.
4	Parity	Indicates a parity error during a read operation. The read will continue, but the V-bit is set to zero.
4 (upgraded MMU)	Read Tape Burst Error	Indicates detection of tape read error >48 bits in length.

MMU Status Word B - Continued

Bit		Description
5	Invalid Manchester	Set if, on receiving data to be written onto memory, the MIA senses an invalid Manchester code. The V-bit is set to zero, the write continues to the end of the block, and the tape stops in the middle of the next subfile.
6	CMD Rcv'd When Not Ready	Set if a command is received while still processing the previous command. If it occurs during writing, the write operation is terminated after the end of the data block, the V-bit is set to zero, and the tape stops in the middle of the next subfile.
7	EOF Block Count ≠ 0	During read or write operations if the block count is not zero on reaching EOF, then the operation is terminated and the unit stops in EOF. The V-bit is set to zero and the BCE will timeout.
8	EOF, Address Not Found	If the commanded address is not found before reaching the EOF, this bit is set. The tape stops in EOF and the V-bit is set to zero.
9	Malfunction	Set due to loss of speed control, loss of phase lock, wrong motor direction, wrong head selection, or a control track parity error. An operation will terminate at the end of the block and the tape stops in the next subfile. The V-bit is set to zero.
10	BOT	Sensing the beginning of the tape causes an uncontrolled stop and the V-bit is set to zero. Power to the unit must be cycled to be reinitialized.
11	EOT	Sensing the end of the tape causes the unit to initialize to BOF 4 and set the V-bit to zero.

MMU Status Word B - Concluded

Bit		Description
12	Action Aborted	Set if an input power transient of from 20 to 250 μs occurs (<15 V pulse and remains <24 V for 20 to 250 μs), if the power supply outputs go out of tolerance, or if speed control is not acquired. Any operation in progress will be terminated and the tape comes to an uncontrolled stop. The V-bit is set to zero and a BCE timeout will occur for read or write operations. If failures occur for other than read or write, the error will not show up until a subsequent command. It is necessary to cycle power to recover from this and cause reinitialization.
13	MIA Bit Count	Set for a MIA bit count error on received data. A write operation will continue to the end of the block and the V-bit is set to zero. The unit will stop in the next subfile.
14	MIA Parity	Set for a MIA parity error on received data. The write operation will continue to the end of the block and the V-bit is set to zero. The tape will stop in the next subfile.
15	Invalid 101 Check	An invalid 101 check on received data causes this bit to be set. The write operation will continue to the end of the block and the V-bit is set to zero. The tape will stop in the next subfile.
16	Data Interrupt	Caused during a write by bits 5, 13, 14, or 15 of Status Word B being set, an MIA address noncompare, or a command sync within 512-word block. The write continues to the end of the block, the V-bit is set to zero, and the tape stops in the next subfile.

5.4 INTERFACING SOFTWARE[4]

The interface between MMU users (applications) and the MMU itself is done by the interfacing software residing in the FCOS. Because of their electro-mechanical nature, resulting in slow access times relative to other BTU's, MMU's require unique I/O software. This software must cover the fixed block length read/writes of 512 words each as well as properly formatting the commands for the MMU. Also, completion monitoring and error handling must be accomplished. Before any commands to an MMU are dispatched, the status of the MMU and BCE are checked for availability. All of these requirements lead to 13 different FCOS programs to handle MMU I/O. Any user of the MMU utilizes these programs to accomplish the I/O.

For an overview of the general flow of the interfacing software, see figure 5-12. This shows only the main flow, and doesn't show any error handling programs that are also part of the operation. The discussion will start at the top (calling level) and progress down to the actual MMU BCE programs.

All requests to use MMU's come into the FCOS programs as Supervisor Calls (SVC's). These are interrupts to the computer process control, each of which has a preestablished priority. This causes the noncyclic MMU processing to be initiated. The SVCs are first processed by the SVC Handler, FPMSVC. This identifies the request and routes it to the appropriate SVC routine to be serviced. Requests for MMU usage can take two paths here. If the request is to support GPC reconfiguration, then the SVC will be passed to MM/GTG_Overlay_Processor (FCMMGPOV). All other requests for use of the MMU will be routed to the I/O_SVC_Service_Processor (FIOSVC).

Special processing must be done on the SVC information coming in for GPC reconfiguration which may involve major function base overlays or operations (ops) overlays, or both. The program FCMMGPOV handles these requests, which come in via the SVC generated by a Merge and Correlate Recorded Output (Program) (MACRO) called OVERLAY (ARC_OVL_PARM).

5.4.1 Reconfiguration

A slight digression is needed here to assure that the reader is not left behind because of unfamiliar terms. There are different kinds of software used in the GPC's to support different parts of the mission. Changing from one type of software to another requires a reloading, or reconfiguration, of the GPC's which is called an overlay. Part of GPC memory is overlayed with new software. The software is further subdivided into resident System Software (SSW), which is always present after GPC initialization (IPL), major function base software, which supports a certain type of processing, and the ops overlay software, which supports a specific part of flight. An overlay can be only the ops type of software, or it may involve both the ops and major function base software. Each piece of software is identified by assigning it a phase number (table 5-I). Each phase is further subdivided into load blocks of related software. When reconfiguration of GPC's is requested, the software can come from either the MMU or another GPC and go to either one or all GPCs. (See section 11.3.2 for details of transitions.)

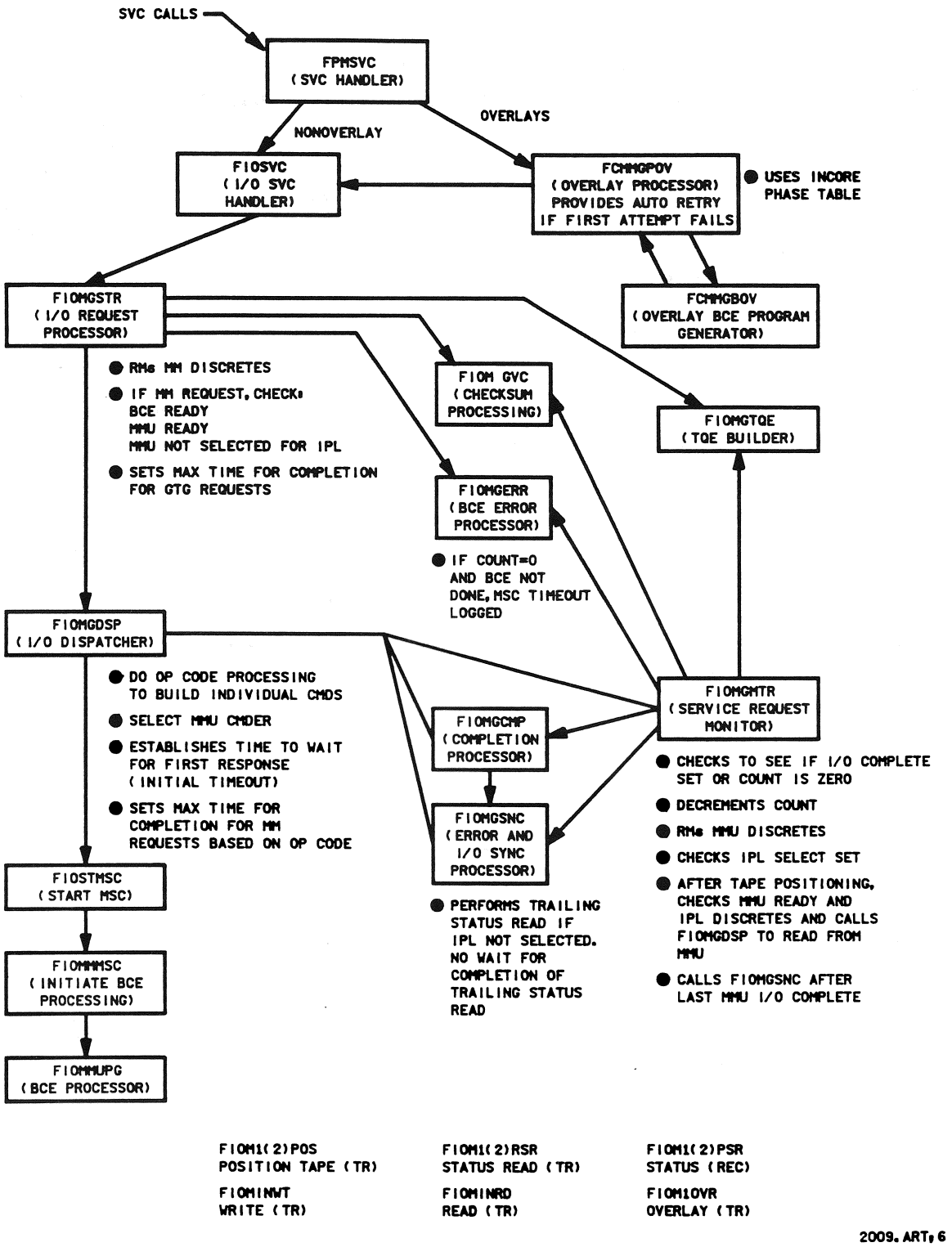


Figure 5-12.- Software flows (only main paths are shown).

TABLE 5-I.- PASS PHASE NUMBER SUMMARY FOR MASS MEMORY[5]

PHASE TABLE (TYPICAL)

Phase no.	Mem. config.	Description
1*	N/A	GPC boostrap loader and mini-directory
2	ALL	Resident system software
3	1,2,3,8,9	GNC major function base (GNC-FB)
4	1	Ascent and abort OPS (GNC 1/6)
5	2	ON-orbit OPS (GNC 2)
6	3	Entry OPS (GNC 3)
7	8	Vehicle utility on-orbit OPS (GNC 8)
8	9	Vehicle utility precount OPS (GNC 9)
9	6,7	Vehicle utility checkout function base (VCO-FB)
10	1	Ascent and abort OPS (GNC 1/6-B)
11	N/A	Reserved
12	6	Vehicle utility mass memory OPS (PL 9)
13	N/A	Reserved
14	4	SM function base (SM-FB)
15	4	Orbit doors OPS (SM 2)
16-20	N/A	Reserved
21	N/A	SM data checkpoint
22	N/A	IMU calibration checkpoint
23	N/A	PDI Decom Format Load (DFL)
24	N/A	PDI Fetch Pointer Memory Load (FPL)
25	N/A	Telemetry format loads
26	N/A	GPC-STP/SW loaders (GPC-STP, SSL, LOAD TBL)
27-32	N/A	Subsystem configuration management lists
33-34	N/A	Reserved
35-44	N/A	SM roll-in displays

*Not currently included in the Mass Memory Phase Table.

5.4.1.1 Overlay MACRO

The MACRO called OVERLAY is invoked in a user program (ARC_GPC_RECONFIG) which also provides parameters describing the reconfiguration. The name of this list of parameters is ARC_OVL PARM. These parameters are:

OVSRCE — This is used to indicate the overlay source; a zero for MM, or a one for a GPC-to-GPC overlay.

OVFNBAS — This is the phase number of the function base overlay if a function base is required.

OPSOV — This is the phase number of the ops overlay being requested.

OVEVENT — This is a flag set to indicate that the overlay request has been completed.

OVSTAT — This is a fullword used to indicate transmitter (source) GPC and receiver (destination) GPC's for GPC-to-GPC overlays. Also, the bus to be used is indicated

	Source GPC					Destination GPC							LDB		MMU bus		
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	18	19	20-31
GPC	1	2	3	4	5	1	2	3	4	5							

Because this MACRO provides only phase information and the MMU knows only file, track, subfile, and block, additional processing is required. This is done by FCMMGPOV using the Incore Phase Table (#PFCMGPT) (fig. 5-13).

Incore_Phase_Table (#PFCMGPT)

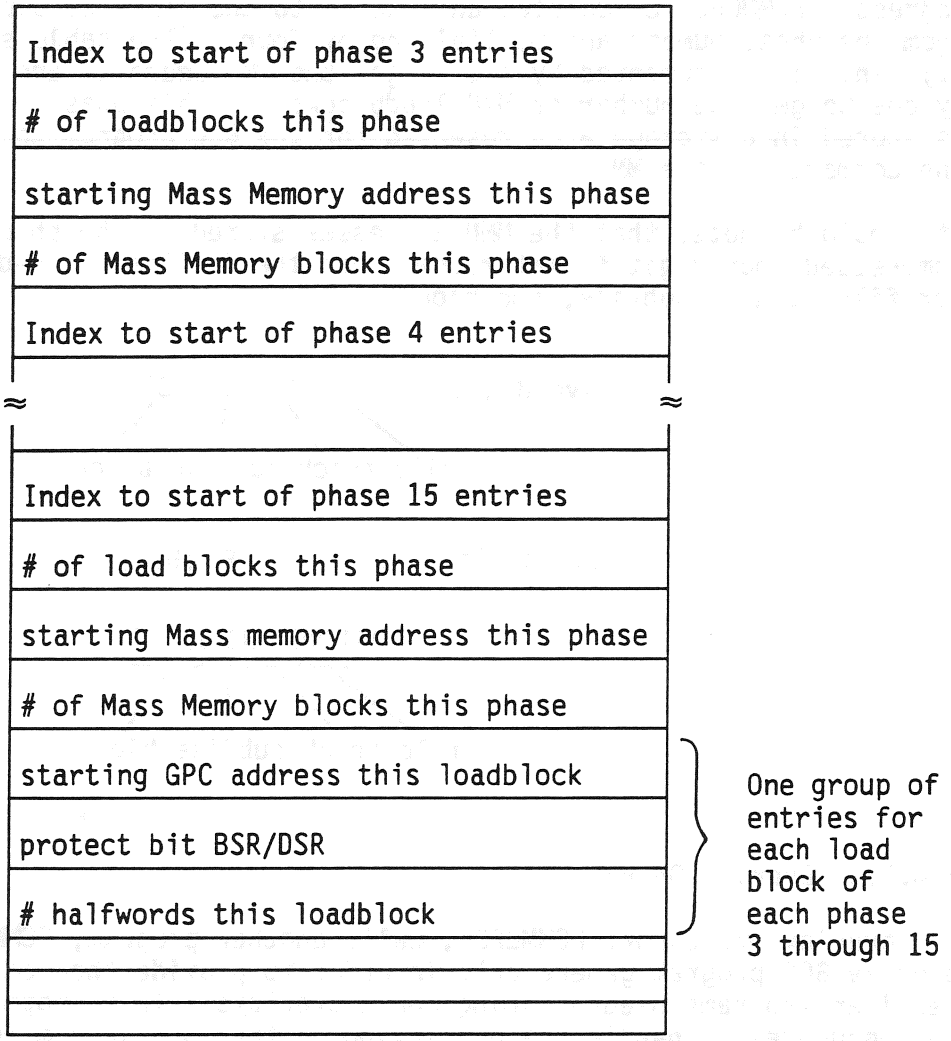
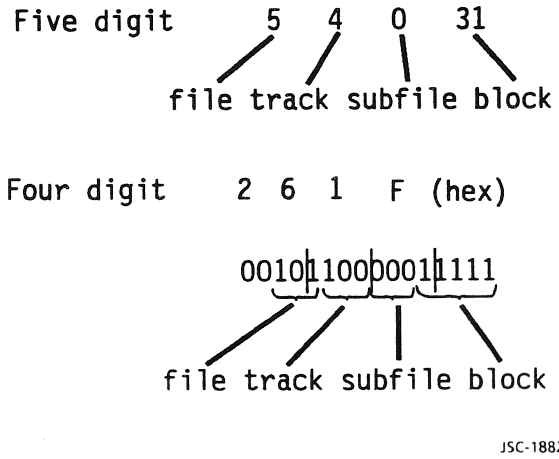


Figure 5-13.- Incore phase table (PASS).

5.4.1.2 Incore Phase Table

The incore phase table provides the link between phase, loadblock, and MMU address. FCMMGPOV calculates an index into the table by subtracting three from the phase number and multiplying by four. (The table starts with phase 3.) This is incremented by two to get the MMU address, and then incremented by one to get the number of MMU loadblocks for this phase. This information is stored in a storage area reserved for the BCE program to use in building the commands to the MMU.

It should be noted that the MMU addresses stored in the phase table are in a compressed four-digit format rather than the familiar five-digit expression for file, track, subfile, and block.



JSC-18820*011

5.4.1.3 Overlay Operations

The overlay processor, FCMMGPOV, calls another program, FCMMGBPV (MM/GTG overlay BCE program generator), in order to provide the real-time BCE receiver programs needed during phase overlays. This program also generates the sequences to handle the overlaying of the data into main memory, as well as the unprotecting and reprotecting of memory. Control is returned to FCMMGPOV.

In the overlay process, it is FCMMGPOV which causes the automatic retry of an overlay transaction if the first attempt fails. This is valid whether the overlay is a GPC-to-GPC (GTG) type or comes from MMU, and is done whether Launch Data Buses (LDB's) or MMU buses are being used.

The next step in the chain of programs is a call to the I/O_SVC_Service_Processor (FIOSVC) by either FCMMGPOV if an overlay, or FPMSVC if other MMU use. FIOSVC initializes the I/O operation and passes the request on to the MM/GTG_I/O_Request_Processor (FIOMGSTR).

5.4.2 Discrete Redundancy Management

The request processor (FIOMGSTR) performs more initialization and then does some preliminary checks before dispatching the I/O request. The first thing it does is call the GPC Discrete_Redundancy_Manager (FCMDSCRM) to process the MM IPL select discretes and MM ready discretes contained in CZ2B_DIA1. This involves checking to see if the majority of common set GPC's (>1/2) agree on the status of these discretes. Only if the majority agree that a discrete has changed from the last agreed-to state is the change entered into the redundancy managed discrete word, CZ2B_DIA_RM1.

Next, the I/O request is checked to see if this is a GTG transaction or an MMU request. If it is a GTG request, then the overlay buffer needed is checked to see if it is available and if not, "MMU OFF/BUSY" is annunciated. For MMU use, the BCE to be used is first checked. If it is busy, "MMU OFF/BUSY" is annunciated. Next, the MMU ready bit in the redundancy managed word is checked. If it is not ready, "MMU OFF/BUSY" is annunciated. Finally the MMU selected for IPL bit is checked in the redundancy managed word. If it is selected for the required MMU, then "MMU OFF/BUSY" is annunciated. If any of these conditions occurs, the processing is terminated with the annunciation of the fault message.

Another part of the processing needs to be explained, but does not really have any effect. The MM/GTG_BCE_ERROR_Processor (FIOMGERR) is called to check the BCE's status to see if any problem occurred the last time that the BCE was used. This would have been for an MMU status read, which is always done at the end of MMU transactions. This trailing status read has no error processing or logging associated with it at the time of occurrence. When the BCE is checked during the start of the next I/O, error processing is done, but no error logging is done here either. The net effect resulting from the error processing is a resetting of the BCE to ready it for the new I/O operation.

5.4.3 Timeout Calculations

If nothing has gone wrong so far, then FIOMGSTR is ready to dispatch the request. Before dispatching the request, a timer count is generated in order to monitor the execution and know the maximum time to wait for the operation to finish. For all MMU operations, except BSR read and position tape, the count is set to 50. This is 50 times 200 ms, or a total possible wait time of 10 sec. For the BSR read case, the count is set to 2 (400 ms), and in the position tape case to 350 (70 sec). If a GTG transfer is requested, then the count is calculated based on the number of loadblocks and total number of words.

For the GTG transfer case, the BCE program execution time is based on 16.5 μs blocks and the number of time blocks is calculated in FCMMGBOV. Each loadblock is considered separately and is subdivided into 2048 word groups, if necessary. The count generated for each loadblock is based on the following expression, where N is the total number of halfwords in a given loadblock, INT is the integer portion of the resulting quotient, and TB is the number of time blocks for this loadblock.

$$TB = 23\left(INT\left(\frac{N}{2048}\right) + 1\right) + 2N$$

The number of counts is calculated as shown below:

$$Count = \left(\frac{TB \times 16.5}{200{,}000} + 1\right)$$

For a typical ops transition involving an ops overlay, a count of 20 is calculated which equates to 4 sec.

After these timeout limits have been calculated, the MM/GTG_I/O_Dispatcher (FIOMGDSP) is called to do further processing of the I/O request and partial building of the MMU command words before the Master Sequence Controller (MSC) is allowed to start the process.

5.4.4 Command Processing

The first thing done by FIOMGDSP is to halt and reenable the BCE to be used for this transaction. This ensures that the BCE registers are clear and that the BCE is reset and ready for use. After this FIOMGDSP splits into GTG requests and MMU requests. This involves building the Program Control Output (PCO) instruction word which will cause the BCE to execute the proper set of instructions.

If the MMU is to be used, additional steps are executed. A tape positioned flag is checked to see if the tape has already been positioned. If not, then an op code of 8 is set to cause positioning to be done. If the tape has already been positioned and the request is for MMU read or write only (not overlay, status read, or positioning), then processing is done to get the address of the I/O buffer in the right format and stored in the BCE register table. Next, MMU op code dependent processing is done for each type of MMU command. This process consists of taking the skeleton command word for each case and OR'ing in the address data of file, track, subfile, block, and number of blocks. This type of information is supplied by either the individual applications programs, or if an overlay, from the FCMMGPOV program which has already been discussed. Now on to the individual command processing.

A. Write CMD Processing

The write processing (FIOMMNWT) starts with the MMU skeleton command word for write enable, which contains the op code '1010.' To this is added the track information. The write command skeleton word has an op code '1000' and a dummy block count of zero. (How or where this gets changed to a nonzero value is not clear.) The track, subfile, and block are 'ORd' into this skeleton word and then it is stored into a BCE command table.

B. Utility CMD Write Processing

The other write processing module is called MM_Utility_Write_Request Processor (FIOMMMUW). It is set up to handle writes of up to 32 blocks of data and is used for capability 1 type writes only (ground commands up the LDB's). This program builds a write enable command word by adding the track address to a write enable skeleton command word. The word count for the transaction is decremented by 1 and then divided by 512 to get a block count to use in the extended block count command word. Next, multiple blocks to be written are handled by generating and storing addresses for each MMU block into the Mass_Memory_Utility_Base_ Register. The write command word is built by adding the track subfile and block information to the skeleton word. The op code is changed to cause a read to verify the write (code 3).

C. Overlay CMD Processing

The processing for op code 7 is done in the MM Overlay Read Request Processor (FIOMMOVL). It builds the extended block count command by using the block count found in TIOQWDCD. Normally this would contain a word count, but for overlay processing, the FCMMGPOV program calculates the block count and places it in this word. Next the read command is built by OR'ing in the track, subfile, and block information found in the variable TIOQSTAD. This comes from the FIOSVC program where the information is transferred from TIOSSTAD which is set in FCMMGPOV where it was read from the incore phase table. Finally, the flag indicating that the last MMU transaction has been completed is set to one.

D. Position Tape CMD Processing

Op code 8 contains the position tape command processing. The command is built by starting with a skeleton word containing the MMU op code for positioning the tape (000). The track, file, and subfile information is obtained from the MM address stored in TIOQSTAD. The track and file are directly OR'd into the command word. The subfile, if nonzero, is decremented by one before being put into the command word. If subfile zero is specified, then the mask for the BOF bit is used instead of the subfile information. This subfile adjustment is done in order to position the tape to the subfile (of BOF) area preceding the subfile to be read from or written to.

5.4.5 MMU Bus/Commander Selection

After the individual op code processing has been completed, FIOMGDSP does MMU commander selection. Although the process is called commander selection, it really sets up a GPC as a listener if it is not assigned to be a commander. If there is no redundant set in existence, then no commander selection is done here. Before commander selection is done, the bus (which dictates the MMU) to be used must be known. The way this information (bus mask word) becomes available to this module is far from clear and therefore warrants an explanation.

There are two ways for the bus masks to be built depending on whether the I/O is an overlay transaction or nonoverlay use of the MMU. For the overlay case, the mask word starts in an applications program called ARC_GPC RECONFIG (GPC reconfiguration). This program sets up the masks and parameters used during a reconfiguration of the GPC's. One of these contains bits to designate the MMU bus to use (if required for the overlay). The word is one of the overlay MACRO parameters ARC_OVL PARM. OVSTAT which is referenced in GPC reconfiguration as ARC_OVL REQ. (The layout of this parameter was discussed in the writeup of the module FCMMGPOV.) The FCOS can then access this parameter, since it is passed across from applications to FCOS as part of the SVC call. The FCMMGPOV program picks up this mask and calls it TOPLSTAT. It then establishes another name, TIOSBUFA. The FIOSVC program picks up the parameter TIOSBUFA and calls it TIOQBUFA. Finally the MM/GTG T/O Request Processor (FIOMGSTR) picks up TIOQBUFA and puts it into the MM BCE mask word (TIOQMNTM).

If the I/O is a nonoverlay use of the MMU, then FIOMGSTR goes to the MMU Major Function (MF) table (CZ2B_MM_MF) to obtain the MMU assigned to the MF doing the I/O. The table contains one halfword for each possible MF (PL, GNC, SM, or ops 0) with bit 3 set if MMU 1 is selected and bit 4 set if MMU 2 is selected. This table is set by item entries (item 1-8) to the DPS UTILITY display or by application program manipulations. The MMU obtained from the appropriate word out of this table is then loaded in TIOMGSTR.

Now that the MM BCE mask word TIOQMNTM has been established, we can press on with the selection of the MMU commander performed by GIOMGDSP. The Nominal Bus Assignment Table (NBAT) is used to determine the commander for this Memory Configuration (MC) for the MMU specified in TIOQMNTM. (The NBAT contains a list of GPC commanders for flight critical strings, CRT's, MMU's, LDB's, and payload buses for each memory configuration.) If the GPC from the NBAT entry is the self GPC (i.e., GPC doing this processing), then self GPC is set as the commander of the bus and no further selection occurs. If the NBAT-specified commander is not the self GPC, then the op code for the transaction is checked. If an overlay or position tape activity is specified by the op code, then default commander selection is done. For this process, if the NBAT specified commander is not in the RS, then the lowest ID GPC of the RS (including self GPC) will be selected as the commander and other GPC's will be listeners.

The default commander selection process was first added to the software in release 18. It was originally intended to be a default commander selection

for all uses of MMU, not just overlay and position tape transactions, the idea being to prevent failing an ops transition because the NBAT-specified GPC had failed out of the set. One problem in having default commander on the overlay read and position type transactions only was with the trailing status read command done as the last command in each MMU transaction. With no default commander selection, if the commander specified by NBAT was not available, the BCE would wait its entire timeout period of 1.96 seconds before timing out, waiting for a response from the MMU. This response never comes because no GPC ever commanded the status read. Because this trailing status read is not monitored for completion, after the command to read status is processed (it thinks it has been started), the software presses on with the next operation. The next operation will find the BCE busy (still waiting) and cause that operation to fail. The proper fix is to make the default commander selection apply to all uses of the MMU. This is planned for release 20 software. The release 19 software has a change in FIOMGDSP that provides an interim fix. The BCE timeout for only the status reads was cut to 330 μs. This is an adequate time for the MMU to return the status words and will not tie up the BCE longer than necessary if the MMU has died or the commander for the read status operation doesn't exist.

Having been buried so deep in FIOMGDSP for several paragraphs, it is now time to step back and get out of it. The main things accomplished were the processing of the individual op codes and the MMU commander selection. After these things are done, the final activity of FIOMGDSP is to do a call to start the MSC processor.

5.4.6 Completion Monitoring

One of the sideline programs which is still important is the MM/GTG IO_Service_Request_Monitor (FIOMGMTR). This program monitors outstanding MMU I/O transactions or GTG transactions to determine if the operation has finished and if so to do completion processing.

If the transaction is a GTG, then the monitoring is fairly straight-forward. If the monitor count (set in FIOMGSTR) is not zero, it is decremented one and checked to see if it is now zero (which means that the operation should be complete and a flag is set indicating I/O complete). Checksum verification is done to check on the validity of the data transferred and if an error occurred, a checksum error is set. The BCE error processor (FIOMGERR) is called to check on any BCE NOGO or MSC timeout that may have occurred. If the I/O complete flag is on because the count reached zero, or the transaction finished, then the flag is reset and the completion processor (FIOMGCMP) is called.

For MMU usage, much more processing is required. First the MMU discretes are redundancy managed by calling FCMDSCRM. Then the MMU ready discrete is checked to see if the MMU is still busy. If it is busy, the IPL select discretes are checked to see if this MMU is selected for supporting on IPL. If it is selected for IPL, then the I/O completion flag is set and an I/O error logged against the transaction attempt, and completion processing is done again.

If the MMU is busy but not selected for IPL, then the wait count is decremented one and checked to see if it has reached zero, indicating that the operation should be finished. The error processor (FIOMGERR) is called to check for BCE problems and a fail/powered down error will be logged and the completion processor called.

If the MMU is found to be ready, then the idle processing is done. If no more I/O transactions are required, the checksum processor FIOMGCV is called to generate and verify the checksum on the transferred data. If a problem occurred, a checksum error will be logged. The BCE error processor (FIOMGERR) is called to check for any BCE problems and then normal completion processing (FIOMGCMP) is done.

If the MMU is ready and more I/O transactions are required, the IPL select discrete is first checked before allowing another transaction to be dispatched. If the IPL select bit is set, then the I/O completion flag is set to force the completion processing to be run. If the IPL select is not set, then FIOMGERR is run to check for BCE errors and FIOMGSNC is run to do error and/or sync processing before dispatching the next I/O transaction.

5.4.7 BCE Programs

Finally, in the main line of programs accomplishing MMU I/O is the Mass_Memory_BCE_Processor (FIOMMUPG). This csect is a collection of the BCE programs needed to read, write, position tape, or read status from an MMU. These BCE programs are not called, but rather they are invoked by having their addresses stored in the MMU BCE program counter by the I/O dispatcher (FIOMGDSP) and then the BCE is started by the MSC processor (FIOMMMSC).

A. Position Tape

The position tape transmitter program reads the status words of an MMU and then sends the prebuilt position tape command.

B. Read Status

The read status transmitter program sends out a request for the status words and then stores them.

C. Status Word Receiver

The status word receiver program is used by listening GPC's to hear the status words read by either a position tape command or a read status command.

D. Write

The write BCE program issues a write enable command followed by a write command. It receives the Status Complete Word (SCW) from the MMU and stores it. It then transmits one MM block of data (512 halfwords) to the MMU and stores the BCE status before returning to a wait state.

E. Read

The read program contains a transmitter section and a receiver section. The transmitter section issues an extended block count followed by a read command. It then branches to the common part of read receiver program. The read receiver program contains delays to get it in step with the transmitter program. After the delays, the common part of the receive program receives the data from the MMU and then stores the BCE status word before returning to a wait state. Note that this read program is for all reads other than overlays.

F. Overlay Read

The overlay read program contains sequences for the transmitter and receiver modes. The transmitter section issues an extended block count followed by a read command. The program then branches to an address in the MMU BCE branch table to execute the dynamically created receive sequences, which are built in the overlay BCE program generators (FCMMGBOV). When the overlay is read into memory, a branch back to the overlay BCE program occurs and the BCE status is read and stored. The overlay read receiver program provides a delay to get it in step with transmitter programs and then branches to the same BCE address tables as the transmitter sequence does.

G. Write Capability 1

There is another MMU write BCE processor called FIOMUWPG which will not be covered. This program is used only for capability 1 writes to the MMU. These are writes to the MMU over the LDB's done on the ground only. These address the MMU by FTSB instead of by phase and loadblocks.

5.4.8 Error/Completion Processing

After an I/O operation completes or times out, completion processing is done by MM/GTG_I/O_Completion_Processor (FIOMGCMP) by calling it from the service request monitor (FIOMGMTR). This may also be called by the I/O request processor (FIOMSTR) if an I/O request cannot be started. All this process does is call error processing (FIOMGSNC) and release resources for other uses in addition to doing cleanup and resetting.

There are a few other FCOS MMU programs that are not mainline programs, but are still important to the operation. One of these programs is the MM/GTG_ MSC_Processor (FIOMMMSC). This program is an MSC routine that initiates

MMU BCE processing. This routine is called by the start MSC_Processor (FIOSTMSC) and exits to the monitor routine (FIOMNTR).

Another of these sideline programs is the MM/GTG_BCF_Error_Processor (FIOMGERR). This program provides BCE and MSC error detection processing. It may be called by the service request monitor (FIOMGMTR) when an I/O transaction completes or the interrupt count expires. It is also called by the I/O request processor (FIOMGSTR) to check the BCE status from the last status read operation.

The other error handling processor is MM/GTG_Error_And/or_Sync_Processor (FIOMGSNC). This program handles processing if errors occur during I/O and ensures that GPC's stay together during the I/O by calling the I/O sync processor (FCMISYNC). If an I/O transaction completes and more transactions are needed to satisfy an I/O request, then the service request monitor (FIOMGMTR) program calls FIOMGSNC. If the last transaction needed to satisfy the I/O has been completed, then the completion processor (FIOMGCMP) calls FIOMGSVC. If errors have been detected, the level C error handler (FIOERRLC) is called to log and annunciate the error. For overlay transactions, if an error occurs, the identity of the GPC is stored in the Transactions Status Word (TSW) along with the Input Program Report (IPR) status. If the transaction is not an overlay, then a transaction failed indicator is set in the TSW. If the transaction is the last to satisfy an I/O request, then FIOMGSNC calls the dispatcher (FIOMGDSP) to initiate an MMU bite status read.

At the end of MMU or GTG I/O operations, a program called MM/GTG_Checksum Processor (FIOMGCV) is called to do checksum generation for each loadblock. This is done by masking out any overflow bits and summing the halfwords. This sum is then compared with the checksum stored in the last halfword of the loadblock. This process is called by the I/O service request monitor (FIOMGMTR).

This has been an attempt to hit the high spots in the MMU FCOS software so that you may be aware of the general flow of data and of where certain key events are performed. Those interested in learning all the details of each software module will find themselves with a long task. Having looked at the hardware and the interfacing software, the next part of this brief will consider the users (applications) of the MMU and how they use it.

5.5 APPLICATIONS SOFTWARE

There are numerous users of mass memory. Some, such as ops transitions, are quite familiar to people while others, such as IMU calibration, are not often considered. There are three ways that applications access mass memory. One is that of having the information hard coded into the program, another is by using an incore table of MMU addresses that is indexed into. A third way is to use a phase table which is either in core or read in from the MMU. A final method involves the use of several of the above types to accomplish an MMU I/O.

The MMU users are discussed in detail in the following sections. Here is an overall summary of the uses of the MMU's and the types of MACRO's they use:

IMU checkpoint:
- Write with checksum
- Read with checksum

Spec 85 MMU R/W:
- Utility write without checksum (the merge operation generates the checksum and adds it to the end of the loadblock)
- Read with checksum

SM checkpoint:
- Utility write without checksum (the checkpoint applications program generates the checksum and adds it to the end of the loadblock)
- Read with checksum

Crew text:
- Utility write without checksum (the merge operation generates the checksum)
- Read with checksum

TFL:
- Read with checksum

Before discussing the MMU users, one more concept which needs explaining is the SM Common Buffer.

5.5.1 SM Common Buffer

The Systems Management (SM) common buffer is an area of memory, in the SM2 and the PL9 memory configurations, which is reserved for the storage of data going to or coming from the MMU's. The effective length of the buffer is 2048 halfwords in SM2 and 16,384 halfwords in PL9. The SM common buffer is comprised of two separate memory areas known as buffer 1 and buffer 2. Buffer 1 is called CDHV_BLOCKS and is in the CVNMMU compool and buffer 2 is called CDIV_RW_BUFR and is in the CDIMMU compool. The SM common buffer is used by:

- SM checkpoints
- Telemetry format loads
- Subsystem Configuration Management (SCM's) (not presently used)
- Mass memory R/W SPEC operations
- Crew text writes via uplink
- Mass Memory (MM) R/W or dumps via uplink

(Note that SM roll-in displays and crew text displays do not use the SM common buffer.)

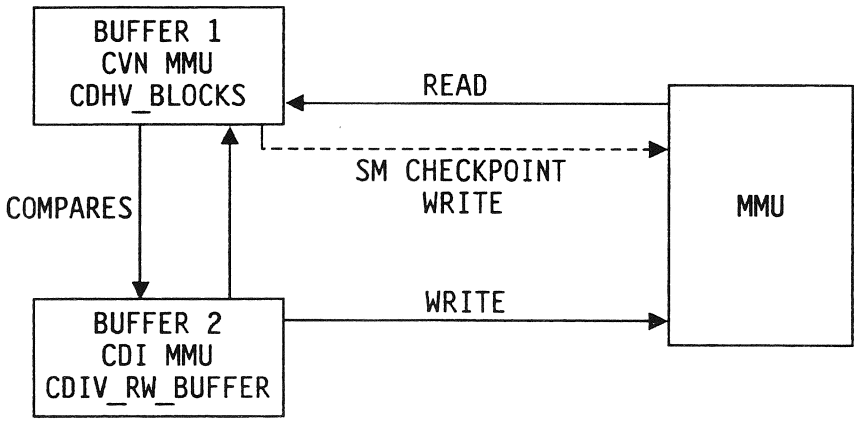
To prevent conflicts by different users of this buffer, there is a flag assigned to the buffer to indicate the availability/nonavailability status to the various users. This flag is called the common buffer busy flag (CDHB_ COM_BUF_INUSE). When any of the above functions are initiated, the flag is set to prevent interruptions by any of the other users, which would cause loss of data. Whenever a user is finished, the flag is cleared.

For crew text writes, MM R/W SPEC writes, or uplink writes to MM, the software logic will clear the flag after completing a write to a single MMU. Another way to clear the flag is to do a Revision Patch Log (RPL) read to either MM, which clears the busy flag. The SM checkpoint write logic also clears the busy flag after completing a write to an MMU.

Note that uplink commands to the MM and R/W SPEC 85 entries are one user of the buffer and do not preclude each other. An MM read and merge could be uplinked, followed by a write done by the crew, all accomplished while the busy flag is set. Or, the ground could uplink multiple writes to only one MM, while the busy flag is set, and not worry about clearing the flag until they were finished with these commands.

Because the SM common buffer is a defined area of memory in the PL9 and SM2 configuration, it is possible to have two SM common buffers, if there is an SM2 GPC and a PL9 GPC. This would also mean that there are two different, independent SM common buffer busy flags (V92X0077X). Presently, the flag is only included in the SM2 downlist. To get the PL9 flag downlisted, it could be put on the OPS0/PL9 variable downlist.

The buffer 1 and buffer 2 areas are used together to accomplish reads, merges, and writes to MM. In general, buffer 1 is used for reads from MM and buffer 2 stores data being written to MM. The exception to this is the SM checkpoint write operation where data are written to MM from buffer 1 instead of buffer 2.



The use of the buffers for a MM read, merge, and write would be as follows:

- Read from MM into buffer 1
- Merge from desired buffer into buffer 1
- Copy from buffer 1 to actual buffer
- Move from buffer 1 to buffer 2
- Write from buffer 2 to MM
- Read back from MM into buffer 1
- Compare buffer 1 to buffer 2
- If dump, dump from buffer 1

5.5.2 MMU Read/Write

The MMU read/write capability is the most generic use of the MMU and provides the ability to access anything on the MMU except for the IPL software. This process is available by a crew display called MASS MEMORY R/W (fig. 5-14), or by ground uplink to the vehicle. The MMU R/W display is only available in the SM ops 2 or Payload 9 (PL 9) memory configurations. In addition, there is the limiting factor at the SM common buffer. In SM2 the buffer length is 2048 halfwords and in PL9 it is 16,384. Because read/writes are done to loadblocks, no loadblock longer than the buffer available may be accessed. For each software release, the loadblocks too long to access will change. If an attempt is made to access a long loadblock, it will be rejected and an error code of 7 displayed.

The display provides the capabilities to read from MMU, write to MMU, dump MMU to ground, and to compare a loadblock in one MMU to the same one in the other MMU. After a specific loadblock is selected, it may be read from either MMU into the GPC buffer. The data can then be dumped via downlist or displayed, 39 words at a time, on the MMU R/W display. When a write is done, the changes are made to the displayed data, it is merged into the buffer, and then written to either MMU.

The two applications programs that handle this read/write capability are VMP_ITEM_PROC and DMP_MM_MSG_PROC. The program VMP is called to do processing of item entries made to the display. The DMP program uses this data or data provided from the uplink processor to build the parameter lists and make the proper MACRO call to initiate the MMU transaction. Some more background on how the R/W display addresses the MMU is necessary before going into the details of either program, but first the details of how to use the read/write display are provided.

5.5.2.1 MMU R/W Instructions

SPEC 85, MASS MEMORY R/W SPEC (fig. 5-14) (available only in SM 2, PL 9 OPS), can be used to perform the following functions:

- MM read
- MM write
- MM dump
- MM 1/MM2 compare

```
9011/C85/          MASS MEMORY R/W          1  273/15:35:02
SM  COMM  BUFF  RDY                            000/01:30:00
DESIRED  ACTUAL    DESIRED  ACTUAL
  1   43AC   53AC     21                        SELECT
  2                   22          MMU  1                 40*
  3                   23          MMU  2                 41
  4                   24          42  S/W  SYSTEM         4
  5                   25          43  PHASE             11
  6                   26          44  LOAD  BLK         12
  7                   27          45  OFFSET       ___45
  8                   28          46  NUMBER            _1
  9                   29          47  PATCH  ID    1111
 10                   30              VIOLATE          NO
 11                   31          READ  LOAD  BLK     48*
 12                   32          MERGE                49
 13                   33          WRITE  LOAD  BLK    50
 14                   34          COMPARE              51
 15                   35          STATUS            CPLT
 16                   36              ERROR  CODE
 17                   37          REGISTER  A
 18                   38                    B
 19                   39          MM  VERSION    19.52
 20                   CHECKSUM    DUMP  VIA  DL       52
```

Figure 5-14.- MMU R/W display.

A. MM Read

1. Select MMU, ITEM 40 (41) EXEC (An asterisk is displayed adjacent to the selected MM.)

2. Set software ID, ITEM 42+_ EXEC

System ID	MM Element
0	Display text and graphics
1	PASS 1
2	PASS 2
3	PASS 3
4	BFS
5	DEU
6	SSME
7	TCS Sequences

3. Set phase, ITEM 43+_ _EXEC

Phase number entered in decimal form. An entry of zero causes the RPL to be displayed. Each time the RPL is called, the MM version line is updated and will display the MM version in decimal. Also, calling the RPL causes the SM COMMON BUFFER BUSY flag to be cleared.

The displayed and actual columns of the data field, ITEMS 1 to 39 and ITEMS 45 and 46 will be blanked when either ITEM 43 or 44 is executed.

4. Set load block, ITEM 44+_ _ EXEC

Decimal loadblock number.

5. Read, ITEM 48 EXEC

Reads the selected MM loadblock into the GPC loadblock buffer. An asterisk will appear adjacent to ITEM 48 and a status word will appear to indicate an in-progress read (IP), a completed read (CPLT), or an error (ERR). The asterisk and status remain until the next entry is made for any ITEM 40 to 52.

NOTE: An MM read will cause the SM COMMON BUFFER BUSY to be set. An RPL read or a write to a single MM is required to clear the flag which releases the buffer.

If an error occurs, an error code appears and remains until the next entry is made for any ITEM 40 to 52.

A reentry is accomplished by a repeat ITEM 48 execution.

Error codes:

1. MM I/O error on read or write.
2. Unresolved phase/loadblock (not found).
3. Configuration control information incorrect or inconsistent.
4. Compare failure.
5. MM checksum failure on read.
6. Phase/loadblock of zero on merge.
7. Loadblock length violation.
8. MM version incorrect.

This error code is useful in determining the success or failure of MM transactions initiated by the crew or by the ground via uplink. The code will remain set until another MM transaction is initiated. It may be placed into the SM variable parameter downlist by calling up to GPC address for the error code's HAL name which is CDHV_PATCH_STAT.

6. Set offset, ITEM 45+_ _ _ _ _ EXEC

Enter offset to determine which 39 words of the loadblock will be displayed. The offset is a decimal number from 0 to 2046 for SM2 and to 16382 for PL 9. For example, an offset of 25 will allow words 26 to 64 to be displayed.

7. Enable display, ITEM 46+_ _EXEC

This is a decimal number from 1 to 39 which will cause the selected 39 words of the loadblock to be displayed under the actual columns. (When used for write operation, this item determines the number of words being changed.)

In addition to the 39 words displayed, several other indicators appear. The checksum is the hexadecimal value from the reading in of the loadblock into the GPC buffer. The register A and B words are the last MM status register A and B contents (hexadecimal). (See section 2.3.4 for the contents of these words.)

B. MM Write

1. Perform steps A(1) to A(6) mentioned in the preceding text under MM Read.

2. Number of changes, ITEM 46+_ _ EXEC

Enter decimal number of words to be changed in the selected loadblock.

This item causes the 39 words, as selected by the offset of ITEM 45, to be displayed in the actual columns of the display.

3. Patch, ID ITEM 47+_ _ _ _EXEC

Patch ID in hexadecimal format is entered here. A nonzero patch ID must be present before requesting a merge in order to avoid an error. No check is performed to detect duplicate patch ID numbers.

4. Enter desired words, ITEM X X + _ _ _ _EXEC

Enter changes by specifying item number (1 to 39) and the hexadecimal word (four digits, including leading zeros) desired for that location. As the entries are made, they will appear in the desired column opposite the actual entries. The number of changes must agree exactly with the value entered for ITEM 46. Also, the changes must be a contiguous block of changes starting at ITEM 1.

5. Merge, ITEM 49 EXEC

This merge entry causes the desired data to overlay the actual in the loadblock buffer, causing the actual to be set equal to the desired.

An asterisk appears next to the merge item number and remains until any ITEM 40 to 52 is entered.

The status indication will indicate an in-progress merge (IP), a completed merge (CPLT), or an error (ERR). (Refer to paragraph A(5) for error codes.)

When CPLT appears, verify that the actual data are the same as the desired data. Additional changes are made by selecting a new offset, number of changes, making the change entries, and then merging.

6. Write to MMU, ITEM 50 EXEC

This causes the GPC to write from the loadblock buffer to the MMU. The loadblock is read back from the MMU after the write has been completed in order for the GPC to perform a word-by-word comparison. An asterisk appears next to ITEM 50 and remains until the next entry is made for any ITEM from 40 to 52. The status indication will indicate an in-progress write (IP), a completed write (CPLT), or an error (ERR). (See step A(5) for error codes.)

The VIOLATE line is a YES/NO indication of the write operation.

NO Indicates that no violations are present.

YES Indicates that a patch ID of zero has been used in the write of a patch or that an unsuccessful patch write was pending when an OPS transition or SPEC termination was processed.

The parameter is updated each time the RPL is read from MM.

If it is desired to write the same data to the alternate MMU, the data are read from the MMU just written to, the other MMU is selected, and then the write is done to the other MMU.

C. MM Dump

1. SPEC 0 PRO

Calls up the GPC MEMORY display.

2. Words/frame, ITEM 42+_ _ EXEC

Specify decimal value of words/frame to be used in downlist for the dump.

3. If in SM 2, SPEC 85 PRO

Return to mass memory R/W.

If in PL 9, RESUME

Return to mass memory R/W.

4. Perform steps A(1) to A(4).

5. DUMP, ITEM 52 EXEC

Dumps selected loadblock via downlist. An asterisk is displayed adjacent to ITEM 52 and will remain until the next entry is made for any ITEM from ITEMS 40 to 52. The status indicator will indicate an in-progress dump (IP), a completed dump (CPLT), or an error (ERR). See step A(5) for error messages.)

D. MM1/MM2 Compare

1. Software ID, ITEM 42+_EXEC

Same as those listed in A(1).

2. Phase number, ITEM 43+_ _EXEC

   Phase number is entered in decimal form. The decimal and actual data columns will go blank as will ITEMS 45 and 46.

3. Load block, ITEM 44+_ _EXEC

   Loadblock is entered as a decimal number.

4. Compare, ITEM 51 EXEC

   This causes a word-by-word comparison of the specified loadblock between MM1 and MM2 to be done by the GPC.

   An asterisk appears next to ITEM 51 and remains until the next entry is made to any of the ITEMS 40 to 52. The status will indicate an in-progress compare (IP), a completed comparison (CPLT), or an error (ERR). (See step A(5) for the error codes.)

The following information will assist in obtaining data from the MASS MEMORY R/W SPEC (SPEC 85, SM OPS 2), via variable parameter downlist capability.

ERROR CODE: This code is useful in determining the success or failure of MM transactions initiated by the crew or MCC via uplink. The code will remain set until another MM transaction is initiated.

GPC ADDRESS: _ _ _ _

HAL NAME: CDHV_PATCH_STAT

CODE

0001 'MM I/O ERROR' on READ or WRITE
- Possible transient
- Recommended - try again

0002 Unresolved PHASE/LOAD BLOCK
- Phase or load block entered incorrectly for entered system ID
- Recommended - correct data and try again

0003 Configuration Control Information Error
- System ID 0 or 7, or...
- All required entries not entered before a read, write, or merge entry, or...
- Incompatible inputs entered for a merge
- Recommended - correct inputs and retry

0004 Compare Failure
- The compared blocks differ in contents between the two MMU's

0005 MM Checksum Failure on READ

0006 PHASE/LOAD BLOCK of ZERO on MERGE
- Not allowed

0007 LOAD BLOCK VIOLATION
- Load blocks too long for the configuration, or...
- Offset too large for the load blocks and patch length
- Recommend - correct data and try again

0008 MM Version Incorrect during MERGE or WRITE

5.5.2.2 Mass Memory Directory

As mentioned earlier, the read/write display allows access to almost everything on an MMU. This is all done through the use of the system ID, phase, and loadblock concept. This was discussed to some extent in the FCOS software discussion in conjunction with ops transitions. The ops transition software is just one part of the MMU software and the incore phase table covers only the ops transition software. To open up the rest of the software, another table called a Mass Memory Directory is used (fig. 5-15).

This directory table divides software into eight major sections, identifying each by a system ID number. Each group of system ID software is then subdivided into phases and loadblocks (see figure 5-16 for a summary of the system/phase divisions). This directory is stored on the MMU in three places. The first one is normally accessed. If a problem occurs in reading this copy, then copy 2 is read and finally, if needed, copy 3 is read. This directory provides either the MMU addresses of the phase data or an MMU address for a phase table that is also read in from the MMU. The phase table would then provide the actual MMU address for the data required.

5.5.2.3 Item Processor

The item processor program handles the inputs made to the R/W display and updates the display to reflect the inputs and display the data retrieved from the MMU. Each setup entry made is placed into a local variable that is passed to the message processor for use in building the parameter lists for the MACRO. In processing the inputs, the item processor uses a string of bits called CDHV_READY_BITS to ensure that all the necessary inputs have been made before accepting a read, write, compare, or dump input item. When one of these items is accepted, then the message processor program will be called to further process the entry and build the MMU MACRO. (See section 5.6.1 for software flow charts.)

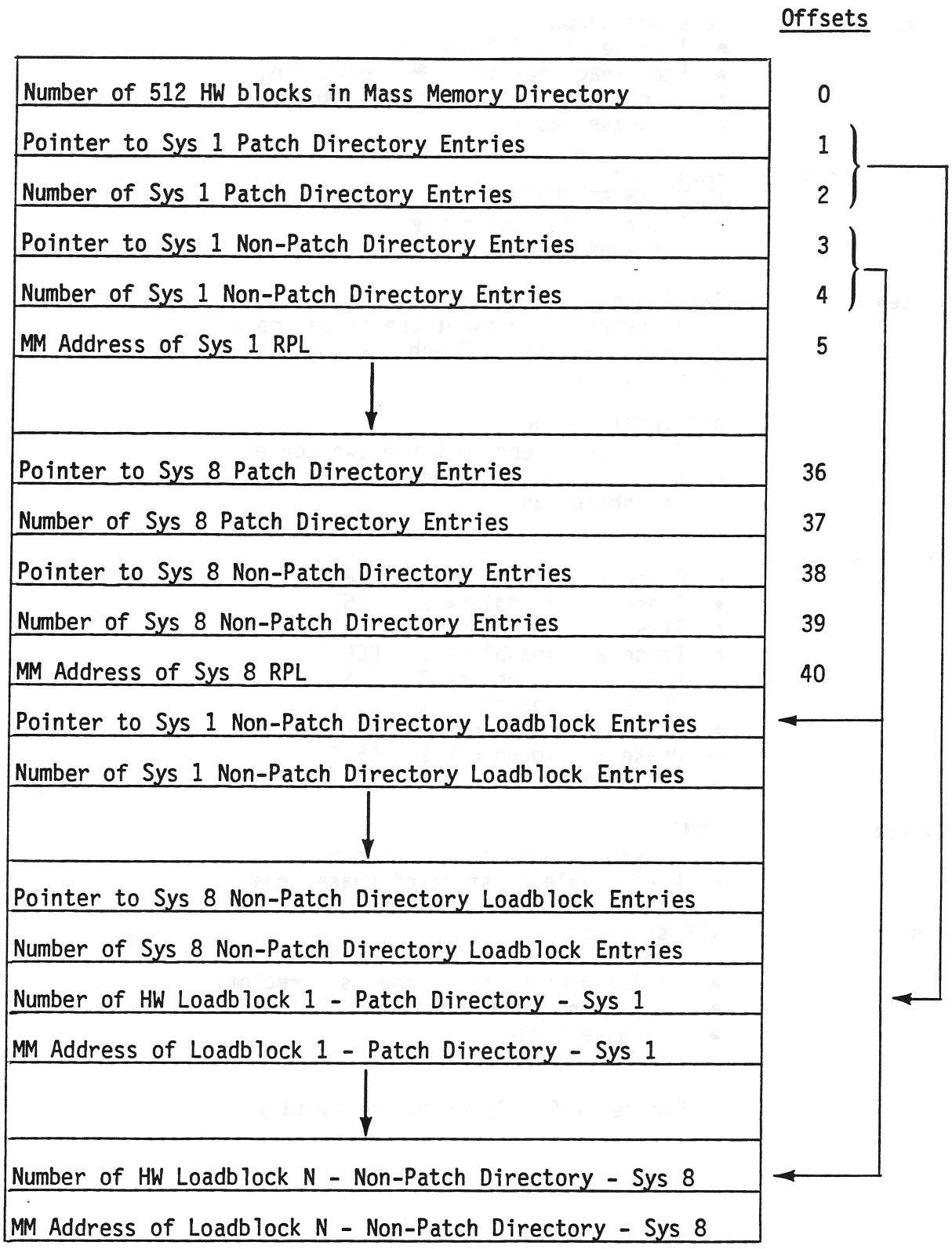


Figure 5-15.- MMU directory.

System 0

Text and Graphics
- 1 phase of 100 loadblocks
- Each loadblock is 2 MM blocks long
- 1 loadblock per crew text display
- No phase table

System 1,2,3

PASS area 1,2,3
- Up to 60 phases
- Phase 1 not accessible
- Has phase table

System 4

BFS (through OI20)
- 3 phases - represent the three copies
- Each phase is 15 loadblocks
- Has phase table

BFS (OI21 and up)
- 2 phases - represent the two copies
- Each phase is 18 loadblocks
- Has phase table

System 5

DEU
- Phase 1 loadblock 1 DCP 1
- Phase 1 loadblock 2 SASTP 1
- Phase 1 loadblock 3 CF 1
- Phase 2 loadblock 1 DCP 2
- Phase 2 loadblock 2 SASTP 2
- Phase 2 loadblock 3 CF 2
- Phase 3 loadblock 1 DCP 3
- Phase 3 loadblock 2 CF 3
- No phase table

System 6

SSME
- 1 phase - represents one copy
- Phase table at start of phase area

System 7

TCS sequences
- 1 phase
- 26 loadblocks, first one is directory
- Subsequent loadblocks are 2 MM loadblocks long
- No phase table

Figure 5-16.- System/phase summary.

5.5.2.4 Message Processor

The message processor can be invoked in two ways. It can be called by the item processor for R/W display inputs, or by the MM uplink processor (VUL_MM_INTERFACE) for R/W's uplinked from the ground. (See section 5.6.2 for software flow charts.) The first thing done by the message processor is to check and see if the SM common buffer is available for use. It does this by checking the buffer busy flag, CDHB_COM_BUF_INUSE. If the flag is not set (i.e., buffer is available) then that flag is set and an internal flag is also set. This internal flag, VMM_COM_BUF_FREE, is used to allow the message processor program to be reentered while the common buffer is left busy in order to complete an MMU transaction. When a transaction is completed, the buffer busy flag can be cleared. If the buffer is busy when the message processor is first invoked, an illegal entry message will be annunciated. Since the buffer is used as the storage area for data going to and from the MMU, proper buffer protection through the use of the busy flag is needed in order to assure the validity of the stored data.

The processing is then broken down into five cases. They are the capability 1 read and writes, and the capability 2 read, merge, and write. In the program these are called MMREAD, MMWRITE, MMLBREAD, MMPMERGE, and MMLBWRITE, respectively. The compare option falls under MMLBREAD as does the dump option. One other item should be taken care of before going on to individual cases. The MMU assigned to this MF (from the DPS UTILITY display) is saved in a local variable called VMM_MF SAVE. This is done so that either MMU may be accessed regardless of the one selected on the DPS UTILITY display. The item entry (or uplink entry) that selects the MMU to be accessed is used to reset the MMU assigned to this MF and will show up as such on the DPS UTILITY display. When operations are complete, the MMU MF assignment is set back to its original state by use of the stored assignment. Now on to the processing done by the individual cases.

The MMLBREAD takes care of read, compare, and dump requests. If a dump request is a non-LDB request and it is a loadblock that was previously read, then a read bypass bit is set that keeps the loadblock from being read again. If the read bypass bit is not set, then the read operation starts.

5.5.2.4.1 MMU directory.- The first part of a read involves reading the MMU directory (already discussed) from the MMU. The address for this directory is taken from a table of addresses called CZIV_MM_ADDR_TBL, which is stored in SSW (table 5-II). The first attempt at reading the directory uses an offset into the table of one, which gives the address of copy 1 of the directory. If this fails, then an index of 12 is used to try copy 2 and finally an index of 13, if needed, to try the third copy. The actual statement that causes the directory to be read is the SVC call, DIO(CDHV_SCK_ RD). This is a replace MACRO for the MMURDSCK MACRO, which is an MMU read with checksum operation.

5.5.2.4.2 Revision patch log.- Once the directory has been read, the RPL will be read in order to obtain the version number of the software stored on the MMU (used for display and to verify validity of a patch). (The RPL contains software version data, a list of patch ID made for this software and a violation flag indicator.) The MMU address of the RPL is obtained from the directory just read by indexing into it. The index is calculated by adding one to the system ID, multiplying by five, and then adding one. (For text and graphics this means that the sixth entry of the table contains the MMU address of the RPL.) If a phase of zero has been requested, the software interprets this as a request to read only the RPL for a given system of software and not the software itself. If only an RPL read is requested, then after it has been read, the SM common buffer busy flag is cleared.

5.5.2.4.3 Read operations.- After reading the RPL, the MM_LOAD_BLK_READ subprocedure is called. This divides into five cases of processing based on system ID in order to know whether it can get an MMU address from the directory, or whether another phase table must be read first.

Text and graphics (sys 0) gets the fourth entry of the table, which is a pointer to the individual loadblock entries in the table. The next halfword after this provides the number of loadblocks. The first entry of the loadblock entries is the number of halfwords in the first loadblock. The next word is the MMU address of this loadblock. Each successive pair of entries takes care of another loadblock. Therefore, by proper indexing into the directory table, the MMU address of the proper loadblock is obtained.

All of the three PASS software copies and the BFS software ID entries are handled by case 2. For these entries, a phase table must be read in from the MMU. The address of the phase table is obtained from the directory by indexing in five times the software ID plus two. This location provides a pointer to the location of the entries for the phase table. This pointer plus two provides the location in the directory where the MMU address of the phase table is located. The phase table (fig. 5-17) is similar to the incore phase table (fig. 5-13) and provides the length and address of each loadblock for each phase.

The phase table is read into the CDHV_BLOCKS buffer (the same one previously used for the directory) from the MMU by the subprocedure called MM_PT_READ. The phase and loadblock requested are checked against entries in the phase table to ensure that they don't exceed the total number of phases and loadblocks for a particular system. If the address information is not contained in the first loadblock of the phase table, then the second loadblock of the phase table is read in from the MMU.

TABLE 5-II.- MMU ADDRESS TABLE CZIV_MM_ADDR_TABL[5]

MMU Element	Table index	Release 20 FACI MMU addresses					
		Allocation 1		Allocation 2		Allocation 3	
	N	F/T/S/BB	HEX	F/T/S/BB	HEX	F/T/S/BB	HEX
MASS MEMORY PATCH DIRECTORY COPY 1	1	44000	2400	CONSTANT	CONSTANT	CONSTANT	CONSTANT
SM CHECKPOINT	2	32000	1A00	22000	1200	12000	0A00
PDI DFL	3	36000	1E00	26000	1600	16000	0E00
IMU CALIBRATION DATA	4	51530	29BE	61530	31BE	71530	39BE
DEU CONTROL PROGRAM	5	44007	2407	44708	24E8	44308	2468
CRITICAL DISPLAY FORMATS	6	44408	2488	44416	2490	44424	2498
SSME	7	03000	0300	CONSTANT	CONSTANT	CONSTANT	CONSTANT
TCS	8	21000	1100	CONSTANT	CONSTANT	CONSTANT	CONSTANT
DISPLAY TEXT AND GRAPHICS	9	31000	1900	CONSTANT	CONSTANT	CONSTANT	CONSTANT
TFL	10	46000	2600	46221	2655	46510	26AA
PDI FPL	11	36304	1E64	26304	1664	16304	0E64
MASS MEMORY PATCH DIRECTORY COPY 2	12	44327	247B	CONSTANT	CONSTANT	CONSTANT	CONSTANT
MASS MEMORY PATCH DIRECTORY COPY 3	13	44725	24F9	CONSTANT	CONSTANT	CONSTANT	CONSTANT
IMU CALIBRATION DATA AREA 1	14	51530	29BE	CONSTANT	CONSTANT	CONSTANT	CONSTANT
IMU CALIBRATION DATA AREA 2	15	61530	31BE	CONSTANT	CONSTANT	CONSTANT	CONSTANT
IMU CALIBRATION DATA AREA 3	16	71530	39BE	CONSTANT	CONSTANT	CONSTANT	CONSTANT
PCS	17	30000	1800	20000	1000	10000	0800
SPARE	18-25	-	-	-	-	-	-

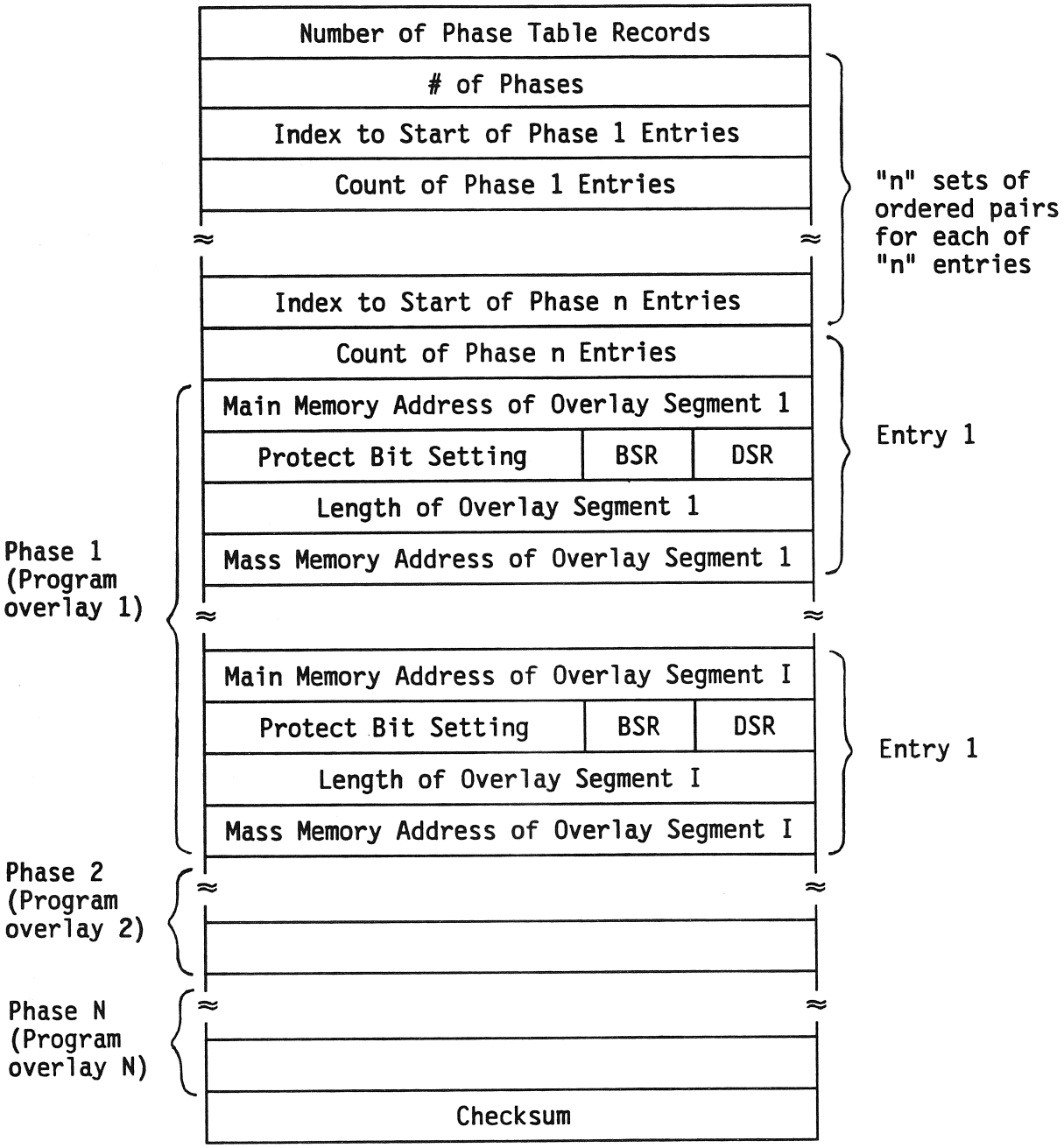


Figure 5-17.- Pass/BFS Phase Table layout.

If the DEU software is requested by entering a system ID of 5, case 3 processing is done by MM_LOAD_BLK_READ. This processing is of similar nature to that done for text and graphics; i.e., there is no phase table to be read in. The addresses required exist in the MMU directory itself. Indexing into the table gets a pointer which leads to another pointer which finally leads to the location where the MMU address is located.

The Space Shuttle Main Engine (SSME) software is handled by case 4 (system ID of 6). This case is like the PASS and BFS processing because an additional table of addresses must be read in from the MMU. This table is called a Main Engine Load Directory (MELD). A separate subprocedure called MM_MELD_READ does the actual reading of the MELD from the MMU into the buffer and then reads the MMU address and loadblock length from this buffer.

The Test Control Sequence (TCS) software is handled by case 5. For the TCS software, the addresses are found in the directory, and no additional table is required.

By the time the individual case processing has been done, an MMU address and loadblock length have been obtained. At this point, the loadblock length is checked against the size of the buffer available (2048 if in SM and 16,384 if in PL 9). If the loadblock exceeds the limit, an error code '7' is displayed on the R/W display and the operation terminates. Otherwise a replace MACRO, DIO(CDHV_SCK_RD), is executed which causes the SVC call to actually read in the desired loadblock from the MMU. (The MMU MACRO is MMURDSCK which is an MMU read with checksum.)

This completes the description of the MM_LOAD_BLK_READ subprocedure. Remember that this procedure had been called from MM_READ_RPL_LB, which had been called under case 3 (MMLBREAD) in the main body of the message processor program. The MM_READ_RPL_LB procedure has one other function, which is to handle the dump inputs made to the R/W display or by the uplinked command.

5.5.2.4.4 Dump operations.- A dump input will cause the read bypass bit to be set, assuming that the dump is of the same area just read from the MMU. With this flag set, MM_READ_RPL_LB bypasses the reading of the RPL and MMU directory, and gets right to the steps for handling the dump. All that is involved here is setting an event called CDWB_MASS_MEM_DMP_REQ. After that, a wait command is issued which causes this program to stop until the CDWE_MASS_MEM_DMP_COMPLETE flag is set. The event flag to start the dump is picked up by the Operational_Data_Processor (DCD_DS2) which causes the MMU loadblock already read from MMU to be placed into the downlist. When this process completes, DCD_DS2 sets the complete flag so that the MM_READ_RPL_LB process knows that the dump is completed and it can then turn off the dump complete flag.

5.5.2.4.5 Compare operations.- Going back to case 3 of the main body of the message processor, there is one other condition it covers. This is the case of an MMU-TO-MMU compare request. For compares the MM_TO_MM_COMPARE subprocedure is called.

This compare procedure starts by assigning MMU 2 to be read from, and calls the MM_READ_RPL_LB procedure to do the reading of the loadblock. The read process completes in the same way that it would for a read request except that control returns to the compare process. If no errors occurred, then the loadblock data are moved from buffer 1 (CDHV_BLOCKS) into buffer 2 (CDIV_RW_BUFR). The MMU 1 is selected and MM_READ_RPL_LB is again called to read the loadblock from MMU 1. If this completes without error, then MM_BUFFER_COMPARE is called to do a word-by-word compare of buffer 1 to buffer 2. If the compare operation is successful, a CPLT indication appears on the R/W display. If any word miscompares, the comparison process stops and an ERR indication is displayed. The final event is to clear the common buffer busy flag.

5.5.2.4.6 Patch operations.- Going back to the main body of the message processor program, case 4 (MMPMERGE) handles the patch requests. This type of request is made after a loadblock has been read and updated. If the request is coming from the R/W display, it means that changes have been entered on the display and exist in a display buffer. If the request is from the ground, then the patches have been uplinked to the uplink buffer, and then moved into the display buffer (CDHV_DESIRED_BUFR). The patch data are checked by MM_CC_VALIDATION to ensure that the patch does apply to the loadblock just read. The uplinked merge command load has an MMU ID in it. However, the message processor software does not look at this MMU ID.

A. Error conditions

In doing the check on the data, the validity processor looks for several conditions. If these are not met, then a specific error code is displayed on the display. The conditions and error codes are shown below.

Condition	Error code if condition not satisfied
MM,SYS,LB, and PHAS all selected and version bit set	3
SYS selected equates to patch SYS ID	3
Phase and loadblock match patch phase and loadblock	3
Version from RPL equals patch version (uplink operations only)	8
Phase and loadblocks other than zero selected	6
For LDB patch, patch count equals no. of words transferred	7
Offset plus patch count doesn't exceed loadblock length	7

If all of these checks are passed, the patch ID ready bit is set and the actual patch operation is started. This is done by moving data from the desired buffer into the common buffer area (CDHV_BLOCKS) by the proper offset. The number and offset ready bits are cleared as the final step.

5.5.2.4.7 Write operations.- Case 5 in the message processor handles the write operations (MMLBWRITE) by calling MM_WRITE_LB. This is for writing a previously read and updated loadblock back to the MMU. One of the first things this process does is to set the patch ID ready bit. This is done to support an MMU-to-MMU copy. The patch ID ready flag is set in the merge operation. If the read is from one MMU and write to the other, there is no merge done and therefore the patch ID ready bit would not get set. The write requires a patch ID to be set and therefore it could not be done. The validity checker routine is called to do the same checks for the write as were done earlier for the merge processing. The write operation itself is broken down into four parts: update the RPL, write the RPL to the MMU, write the loadblock to the MMU, and write to the RPL.

Updating the RPL consists of putting in the new patch ID, incrementing the patch count, and setting the violation indication to assume an error occurs during the write. A new checksum is generated for the RPL and then it is written to the MMU.

The next step is to do the actual loadblock write to the MMU. The data are written to the MMU from buffer 2 (CDIV_RW_BUFR), using the MMURMUM MACRO (write without checksum). The FCOS program FIOMGDSP causes a reading back of this data from the MMU after the write completes in order to verify the write. The data are read into buffer 1 (CDHV_BLOCKS). Then the compare process is called to do a word-to-word comparison between buffer 1 and 2. If the comparison finds no error, then the final section of the write is to write the RPL to the MMU again. This is done to turn off the violation flag. Finally, the common buffer busy flag is cleared and the write operation is finished.

This concludes the cases of the message processor but it has one final operation. It restores the MMU assignment for this MF back to what was originally assigned on the DPS UTILITY display.

5.5.2.4.8 Ready bits.- At the beginning of the discussion about the R/W operations, some bits called the ready bits (CDHV_READY_BITS) were mentioned. These bits have come up from time to time during the discussions of the individual operations, but a summary of where these bits are set and checked is really needed to keep the sequencing in perspective. The following table shows when bits are set (S), cleared (C), or required (R) for an operation to proceed.

Item processor ↓	MM	SYS	LB	PHS	OFF	NUM	PTH	VER
Item 40 MMU 1	S							
41 MMU 2	S							
42 System ID		S						
43 Phase				S				
44 Loadblock			S		C	C	C	C
45 Offset					S			R
46 Number			R	R	R	S		
47 Patch ID							S	
48 Read	R	R	R	R	C	C	C	
49 Merge	R	R	R	R	R	R	R	
50 Write	R	R	R	R			R	R
51 Compare		R	R	R				
52 Dump	R	R	R					
Message processor ↓	**MM**	**SYS**	**LB**	**PHS**	**OFF**	**NUM**	**PTH**	**VER**
Capability 1 write	C	C	C	C	C	C	C	C
Capability 1 read	C	C	C	C	C	C	C	C
MM_READ_RPL_LB	S	S						S
MM_LOAD_BLKREAD (invalid phase or loadblock entered)			C	C				
(loadblock too large)			C	C				
(correct entries)			S	S				
MM_PATCH_MERGE							S	
(invalid input)					C	C		
MM_CC_VALIDATION	R	R	R	R				R
MM_WRITE_LB (1st operation)							S	
(2nd operation)							R	
(3rd) after write cmplt							C	C
MM_PT_READ (if phase or loadblock error)			C	C				
MM_MELD_READ (if phase or loadblock error)			C	C				

5.5.2.4.9 Use of SM common buffer during R/W operations.- Another point that needs a summary is the use of the SM common buffer by the R/W operation. The buffer busy flag is set at the beginning of the message processor program the first time it is called, if the buffer is not already busy. There are only three ways that the buffer flag is cleared in the message processor: an RPL only read, a write to an MMU, and an MMU-to-MMU compare operation. For any other operation, the flag will remain in the busy state, protecting data in the common buffer from being destroyed by other users. If SPEC 85 is dropped, cleanup processing will clear the buffer.

5.5.2.4.10 Uplink procedure for MMU patch operations.- Uplink writes to a single MMU require an uplink read followed by an uplink merge, and finally an uplink write. The MMU should go busy twice for the read, stay ready during the merge, and then go busy four times during the write operation. The total number of ready/busy cycles may not be observed on the ground due to the timing involved. The SM common buffer should go busy at the start of the read operation and stay busy until the write is completed.

If it is desired to write the same data to the other MMU, there are two ways it can be done. An entire read, merge, write sequence could be done, but it is not necessary. The shortest way is to do a read of the MMU already written to, followed by a write to the other MMU. This will satisfy all the built-in software checks on the validity of the operations. The lengthy merge process is avoided. For this read/write operation, the SM common buffer should go busy at the start of the read and stay busy until the write operation is complete.

If a write to the second MMU is attempted without the preceding read, the operations will fail and an MMU error code of 3 would be expected. In addition, the SM common buffer would be locked up in the busy mode.

This concludes the discussion of the MMU R/W display/uplink processors.

5.5.3 OPS Transitions

The rest of the MMU users can be divided into three categories: ops transitions, MMU address table users, and roll-in display users.

Ops transitions is one of the high visibility users of the MMU's. Although the whole ops transition process involves a lot of different interactions and highly complex software, all the calls for MMU use come from one program called ARC_GPC_REC. This reconfiguration type program can call for MMU's to support either a prepositioning activity or an overlay read operation.

The prepositioning activity is invoked by using a replace MACRO ARC_MMU1(2)_POS which is a MMUPOS MACRO that generates the SVC call to the FCOS software. The reconfiguration program gets the address from the incore phase table and passes this along with the SVC call. The MMU that the position request is directed to is driven by the MMU MF assignment for this MF. The reconfiguration program sets this to the desired MMU before issuing the SVC call. The FCOS programs (discussed in the interface section of this brief)

pick up the MMU assignment information in order to cause the proper MMU to be driven.

Overlay read requests for MMU use are made in a subprocess called ARC_MEM_OVL. The replace MACRO used is ARC_OVL_PARM which is the OVLYLST MACRO already described in the interface software discussions.

5.5.4 SM Checkpoint

The MMU address table (table 5-II) provides starting addresses for many users of the MMU. Entries number 1, 12, and 13 in the table have already been discussed in the R/W capabilities section. Entry number 2 is for the SM checkpoint operation.

The SM checkpoint operation provides the capability to store the current status of a group of predefined SM parameters and the current time onto the MMU and then retrieve them when required to reestablish a given configuration in a newly IPL'd SM GPC. The checkpoint store operation is started in a program called STM_TABLE_MAIN. This program will reserve the SM common buffer by setting the buffer busy flag if the buffer is not in use.

The program then calls SCK_PNT which collects the data for the checkpoint and generates an SVC call through a MACRO to cause the MMU write operation.

The MACRO is called CSZV_DATA_LWRT, which is a replace MACRO for MMURMUW, which is an MMU utility write. The address information for the MACRO comes from indexing into the MMU address table. Before this is invoked, a checksum is generated and stored with the checkpoint data going to the MMU. After the write operation completes, whether successful or not, the SM common buffer is released by clearing the buffer busy flag.

The retrieval of the checkpoint information can be invoked during the initialization of the SM ops by a program called S21CLNUP. If the checkpoint retrieve enable flag on the DPS UTILITY display is set, or if no checkpoint time is in memory, then a call is made to SRE_STORE to read the information from the MMU. This program will turn off the checkpoint retrieve flag if it is set and then go to the MMU address table to get the checkpoint address. It invokes the MACRO CSZV_DATA_RD which is a replace MACRO for MMURRDCK (remote read with checksum), which causes the SVC call to read the checkpoint data from an MMU into the common buffer. If the checkpoint retrieve flag has been set, then the checkpoint time is displayed and the data re moved to the proper areas in memory. If the checkpoint retrieve has not been enabled, then only the time is displayed and the data are not used. It should be noted that the retrieve operation doesn't use the common buffer busy flag. Because this retrieve operation is done as part of SM initialization, there are no other users of the buffer around to interfere with the retrieve operation.

5.5.5 Crew Text

Another user of the MMU address table is the display text and graphics also known as crew text or TUMS. These display pages (numbered from 900 to 999) are handled by the program DXR_PAGE_TST. Each display page is 1024 words in length and requires two MMU blocks for storage. The address of a particular page is calculated by dropping the hundreds digit from the display number, multiplying by two, then adding the starting address found in the address table's ninth entry. The call to an MMU is issued in DXR_DMM_ROLLIN. It uses a MACRO called DXRV_MMRD_PLIST, which is a replace MACRO for the MACRO MMURRDCK which causes the SVC call to FCOS. The operation is a remote read with checksum.

5.5.6 Roll-In Displays

A user of MMU that is processed similarly is the roll-in type display operation. The MMU addresses for these displays are stored in a display format buffer 6 csect called #PCDR15. The same module that the crew text processing uses, DXR_PAGE_TST, is used to read the display buffer area to obtain the MMU address. Then DXR_DMM_ROLLIN issues the SVC call in the same way it does for a crew text display.

The rest of the MMU address table users work in similar fashion. Their application programs index into the table to get MMU addresses. They are used as absolute addresses or as bases to add to in order to get the actual MMU address required.

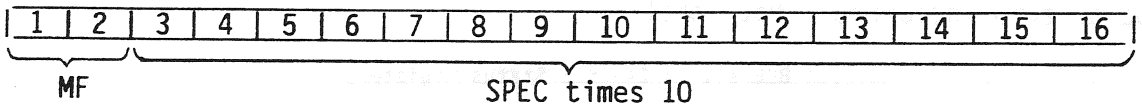
DXR_DMM_ROLLIN is the process that handles the roll-in requests. When a roll-in request is received, a check is made to see if I/O is already active for another request. If another roll-in request is already active, and the new request is not the same as the request in progress, then the new request is terminated and an "OFF/BUSY MMU" fault message is annunciated. No attempt will be made to actually dispatch the request to the MMU, so no I/O error log entry will be made.

If no other roll-in request is currently being serviced, then a check is made to see if that spec is already available in one of the two roll-in buffers (DFB4 and DFB5). If the Spec is not found, buffer availability is checked. If a buffer is found that is not in use, a flag is set to prevent that buffer from being used for display. The MMU is then accessed, and when the transaction is completed without error, the flag will be reset and the buffer contents will be displayed. If an error is encountered, the MMU error page will be displayed and the flag will not be reset. This means that a Spec may be resident in the buffer but cannot be displayed.

If there is a question concerning what is actually in the buffers, the following memory addresses may be interrogated:

DFB4 - #PCDG154
DFB5 - #PCDH155

The tenth entry in these compools contains the major function and the Spec number times 10.



This has been a summary of the MMU users. It is not intended to give every detail of each program involved, but rather to convey some of the significant points that are asked about from time to time. Hopefully, this overview will be useful to anyone doing indepth study of the MMU software.

5.6 MMU ERROR SIGNATURES

5.6.1 I/O Background

Errors involving the MMU are characterized by a combination of the following components:

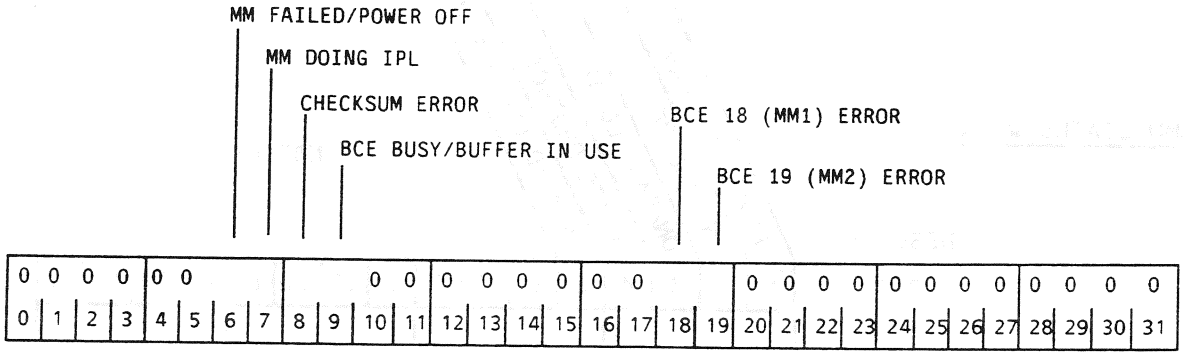
'I/O ERROR MMU'
'MMU OFF/BSY'
BCE 18, 19, 27 or 28
MMU STATUS REGISTERS A & B

In order to determine which component will apply to which type of error condition, the following background notes will be useful.

5.6.1.1 BCE 27

Before the GPC initiates a command, it checks to see if the MMU is OFF/BUSY, selected for IPL, or the BCE is busy. These types of errors are annunciated by the BCE 27 error log which primarily indicates errors that are external to the MMU. It is also possible to get a BCE 27 after the completion of a transaction but only to indicate a checksum error.

BCE 27 (Mass Memory Status Register)

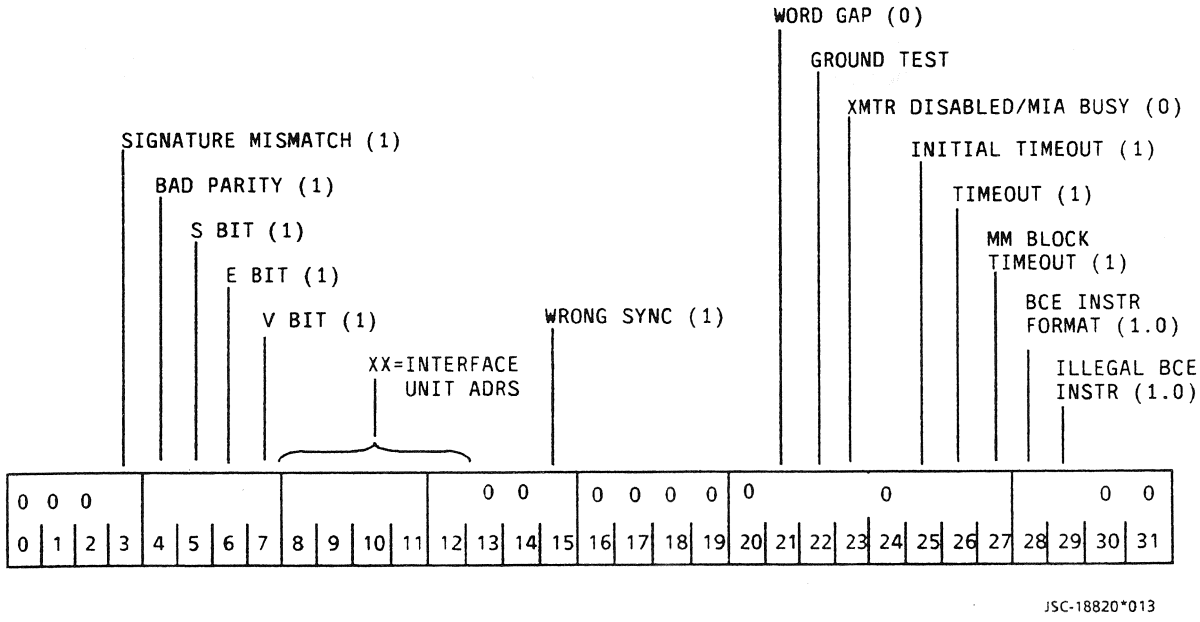


JSC-18820*012

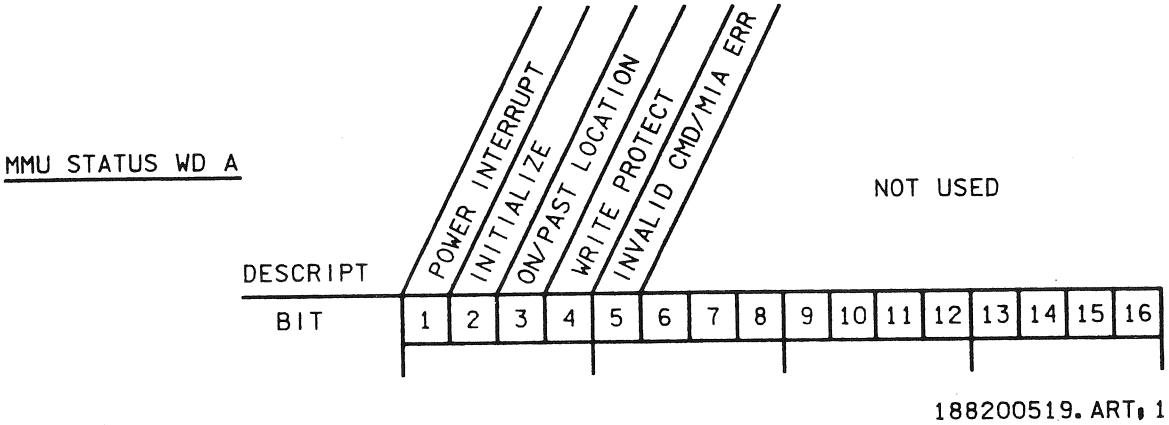
5.6.1.2 BCE 18 or 19

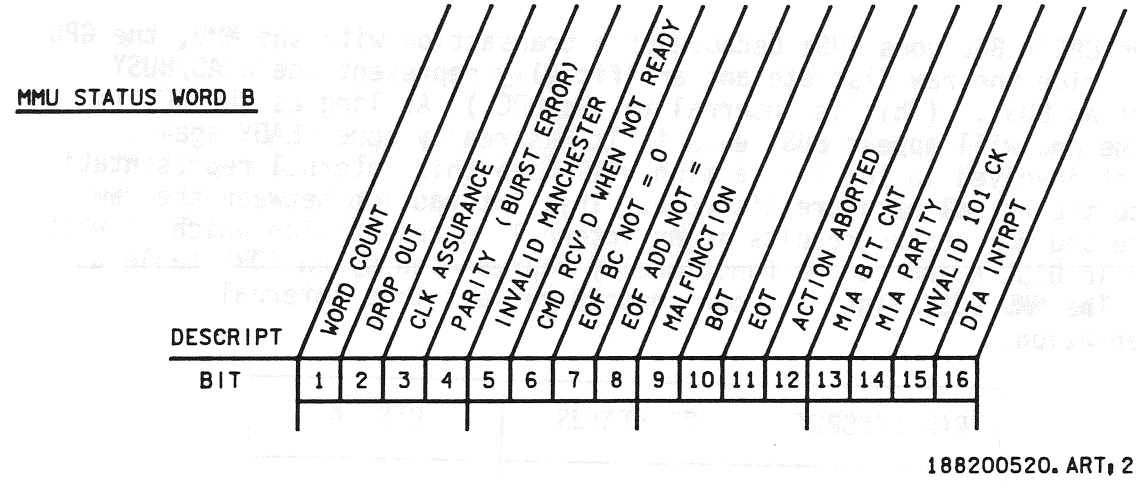
BCE 18 or 19 is logged when the MSC monitor decides that the transaction has completed, but a BCE NO-GO was detected. The stat word will indicate what type of error occurred. Note that

(a) BCE's 1 to 24 (BCE Status Register)



the V-bit is indicated in this word. The MMU will reset the V-bit to zero when certain MMU STATUS A & B bits are set. The V-bit is reset if an error is detected on the current transaction. The only way the GPC will see the V-bit is if the MMU is capable of sending data from the current transaction back to the GPC. If the GPC requests the STATUS register contents from the MMU, this is a separate transaction and the GPC always sets the SEV bits to the nominal '101' pattern before reading the STAT registers, thereby erasing the V-0 that was set on the previous transaction. Therefore, even if you know that a V-bit had been reset, you may not see it in the error signature. This is the case for the access time error signature which is discussed in section 5.6.3.



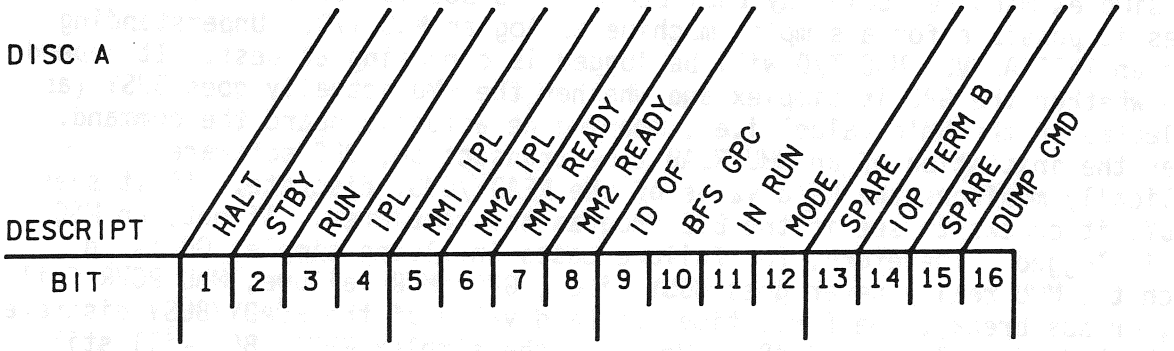


188200520.ART, 2

One of the typical BCE NOGO cases is the INITIAL T/O which is discussed in section 2.5.1.3.

5.6.1.3 Ready Discretes and Timeouts

Each GPC generates its version of DISCRETE WORD A. In particular, each GPC's perception of each MMU's READY/BSY status is generated as a combination of two things: the raw hardware discrete from the MMU; and whether the GPC's BCE is BUSY with a transaction involving the MMU.

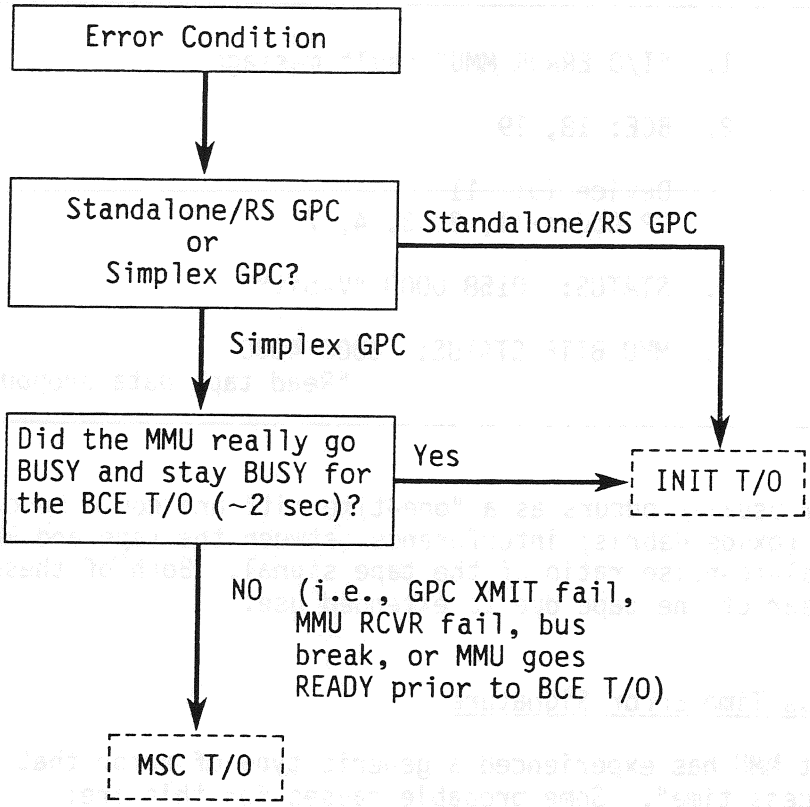


188200521.ART, 2

When the GPC's BCE goes BUSY because of a transaction with the MMU, the GPC will override the raw discrete and artificially represent the READ/BUSY discrete as BUSY. (This is internal to the GPC.) As long as the BCE is BUSY, the MMU will appear BUSY even if it has really gone READY again. GPC's not involved in the transaction will have this internal representation equal to the actual hardware discrete. This interaction between the raw discrete and BCE status results in an internal representation which is what is seen in DISC A and can be functionally represented by an "OR" table as below. The MMU BUSY lights on our console reflect this internal representation.

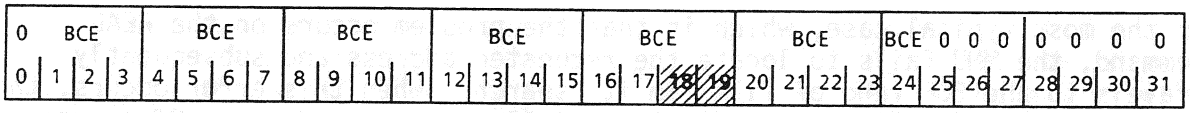
RAW DISCRETE	BCE STATUS	DISC A
READY	READY	READY
READY	BUSY	BUSY
BUSY	READY	BUSY
BUSY	BUSY	BUSY

Now let's consider the redundancy management of READY discretes. It is this DISC A opinion from each GPC that is RM'd. This means that, if the GPC is a standalone or RS machine, the RM'd value of DISC A will say BUSY as long as the BCE(s) is (are) BUSY. For a simplex machine, however, the RM'd value from the other CS members will accurately reflect the MMU raw discretes and the simplex machine could say BUSY because his BCE is BUSY. This conflict makes it possible for a simplex machine to log an MSC T/O. Understanding when an INITIAL VS. MSC T/O will be logged is confusing at best. It depends upon whether the GPC is simplex and whether the MMU actually goes BUSY (as reflected in the RM'd value) i.e., meaning he actually heard the command. After the initiation of an MMU READ type transaction, GPC software cyclically monitors the RM'd value of the READY/BUSY discrete. If it sees READY, it checks to see if the BCE program has completed. If not, an MSC T/O is logged. Therefore, for failure cases involving simplex GPC's in which the MMU really never goes BUSY (i.e., GPC XMTR failure, MMU RCVR failure, or bus break), the first time the RM'd value of the READY/BUSY discrete is checked, it will show READY. However, the simplex GPC's BCE will still be BUSY, so an MSC T/O will be logged. The following flowchart may be useful in determining INIT VS MSC-T/O.



(c) BCE 28 (MSC Timeout)

0000	2000	18	MM 1	MU 1
0000	1000	19	MM 2	MU 2



JSC-18820*014

5.6.2 Data Dropout Errors

In addition to the access time error signature (which will be discussed in section 5.6.3), another typical MMU error is a data dropout. The error signature is given below and is supported by KSC hardcopies.

1. "I/O ERROR MMU" fault message

2. BCE: 18, 19

   Device ID: 11
   OP Code: 1, 2, 3, 4, 7

3. STATUS: 0158 0000 "V-bit"

4. MMU BITE STATUS: 0000 4000
   "Read tape data dropout"

This problem usually occurs as a "one-time hit" and could be caused by particulate (oxide debris) interference between the tape and READ head, or by low signal-to-noise ratio of the tape signal. Both of these problems are caused by wear of the tape due to extended use.

5.6.3 Access Time Error Signature

Every flight MMU has experienced a generic type of error that has been labeled "access time". Some probable causes for this are:

A. An oxide buildup on the tape

B. Oxide debris between the read head and control track

C. Tape failure to come up to speed within an appropriate time period because of incorrect tape tension, roller slippage, motor noise, or motor brush wear.

In the most typical case, which is that the problem occurs on the READ command, the MMU fails to locate the requested address and subsequently travels to the EOF (end of file) in its search. When this error occurs, the transaction is usually completed successfully on a second try. Of course, if it happens during an OPS transition, there is an automatic retry.

Based upon KSC hardcopies, the following typical error signature has been isolated. In all cases except one, this signature will be annunciated by any GPC in the transaction, regardless of whether it is a simplex, redundant set, or standalone GPC. Also, all RS computers will log the same error. (This is the expected signature if the access time problem occurs on the READ or WRITE command.) (One special case has been analyzed in which the simplex machine will annunciate an MSC T/O. See table 5-III.)

TABLE 5-III.- ACCESS TIME ERROR SIGNATURE ON READ OPERATION

Command/transaction	MMU Response	RS or standalone GPC response	Simplex GPC (i.e., SM) ①
② 1. STATUS REQUEST	Transfers BITE register then goes READY. (No access time error possible since does not involve tape movement.)	GPC waits for status return before sending PTC. (contents of status are not considered when sending PTC.) If status does not return, logs BCE 18 or 19 INIT T/O and does not send PTC.	Same
② 2. POSITION TAPE CMD (PTC)	Due to the time involved for the tape to come up to speed, all position commands move the tape to the middle of the subfile preceding the subfile to be accessed. If an access time problem occurs on the PTC (1a and 1b below), MMU could travel to BOT or EOT in search of the address. Would set BITE bit 10 or 11 and V-bit. (MMU does not return data to the GPC.)	Waits up to 70 seconds then sends READ cmd. If MMU does not find the address, GPC is unaware since it does not receive data back on a PTC.	Same

Notes: 1. "Standalone" means it is the only GPC up and running. "Simplex" means it is a CS minority.

2. These commands are chained and are included in one BCE program.

TABLE 5-III.- Concluded

Command/transaction	MMU response	RS or standalone GPC response	Simplex GPC (i.e., SM) ③
3. READ COMMAND	Case 1: Access time problem on PTC.		
	a. MMU is still BUSY traveling to EOT or BOT. Sets BITE B bit 6.	a. IF MMU still BUSY when GPC tries to send READ command, will log BCE 27.	Same
	b. MMU incorrectly positioned itself on PTC. When READ cmd is received, MMU detects incorrect address compare, sets BITE A bit 2 and V-bit, reinitializes electronics to "forget" this cmd, then goes READY.	b. GPC sends READ command, sets up BCE timer and waits for data. Logs INIT T/O when data does not return. (As long as BCE is BUSY, GPC sets internal MMU BSY discrete even if MMU really is READY. The RM'd discrete says BUSY, too.) Don't see the V-bit because data did not return.	Logs MSC T/O because GPC sets MMU BUSY internally as long as BCE is BSY, but the RM'd value of the RDY discrete says READY as an accurate reflection of MMU status. Do not see V-bit.
③	Case 2: Access time problem occurs on READ cmd.		
	When the READ cmd is received, the tape starts moving and is up to speed by the time it reaches the subfile where the reading is to be done. If access time problem occurs, will go to EOF looking for address of start of data. Sets BITE B bits 7 and 8 and V-bit and goes READY. See BITE bits on trailing STATUS REQUEST.	GPC sends READ cmd, sets up BCE timer and waits for data. Logs BCE 18 or 19 and INIT T/O when no data returns because of access time problem. Do not see the V-bit because data did not return.	Same - because the RM'd value of the discrete says BUSY because the MMU actually went BUSY.
4. STATUS REQUEST	Transfers BITE contents then goes READY.	GPC receives BITE and downlists. GPC does not perform error checking on the BITE.	Same

Notes: 1. "Standalone" means it is the only GPC up and running. "Simplex" means it is a CS minority.

2. These commands are chained and are included in one BCE program.

3. These commands are chained, but are two separate BCE programs.

TABLE 5-IV.- TYPICAL ACCESS TIME ERROR SIGNATURE

Access Time Error Signature:	1. "I/O Error MMU" fault message 2. BCE: 18, 19 DEVICE ID: II OP CODE: 1. Write w/Checksum 2. Write w/o " 3. Read w/ " 4. Read w/o " 7. Read OPS O/L 3. STATUS: 0000 0040 "INIT T/O" 4. MMU BITE STATUS A = 0000 B = 0300 "EOF Block Count = 0" "EOF Address Net Found"

Notice that a V-bit was not annunciated. The reason for this was explained in section 2.5.1.2. Also notice that the timeout is an INITIAL, not an MSC. A detailed discussion of timeouts was presented in section 5.6.1.3.

Table 5-III attempts to present a "roadmap" for what happens during a READ operation and what the MMU and GPC response is if the access time problem occurs at certain points during the operation. Even though this table presents a READ operation, the same responses and error signature should apply to a WRITE operation as well. By analysis, it is determined that the access time problem could happen on the POSITION TAPE COMMAND (case 1 a and b) or on the READ command (case 2). Case 2 is the typical case and is supported by KSC hardcopies.

5.7 REFERENCES

1. Shuttle Operational Data Book, JSC-08934, November 3, 1989.

2. MC615-0005 MASS MEMORY ORBITER, May 5, 1977.

3. Operation and Maintenance Instructions, Odetics Inc., NAS-9-14000, September 16, 1976.

4. OI A Software, 1988.

5. MMU Program Release Notice, JSC-16764, MPRN OI 8C, October 5, 1989.

5.8 MMU SOFTWARE FLOWS

MASTER
TIMING
UNIT


MASTER
TIMING
UNIT

SECTION 6
MASTER TIMING UNIT

CONTENTS

Section		Page
6.1	INTRODUCTION	6-1
6.2	DESCRIPTION	6-1
6.2.1	Operational Constraints	6-1
6.2.2	Connectors	6-4
6.2.3	Pin Definitions	6-5
6.3	POWER	6-8
6.4	OSCILLATORS	6-8
6.4.1	Select Logic	6-9
6.5	MTU SIGNAL DISTRIBUTION	6-10
6.5.1	Power Divider/Frequency Divider	6-11
6.6	ACCUMULATORS	6-12
6.6.1	Accumulator Data Content	6-12
6.7	ACCUMULATOR TIME OUTPUTS	6-13
6.7.1	FF MDMs	6-13
6.7.2	OF1 MDM	6-14
6.7.3	OF2 MDM	6-14
6.7.4	Interrange Instrumentation Group B (IRIG B)	6-14
6.8	UPDATE CIRCUITRY	6-16
6.8.1	Update Word Format	6-17
6.8.2	Update Circuitry Overview	6-18
6.8.3	Update Logic	6-19
6.9	BITE STATUS REGISTERS	6-20
6.9.1	Bit Definitions	6-21
6.10	ANALYSIS OF THE BITE WORD	6-22
6.10.1	BITE Bit 1	6-22
6.10.2	BITE Bit 2/3	6-22
6.10.3	BITE Bit 4	6-22
6.10.4	BITE Bit 5/6	6-22
6.10.5	BITE Bit 7/8	6-23
6.10.6	BITE Bit 9/10/11	6-23
6.10.7	BITE Bit 12	6-23
6.10.8	BITE Bit 13/14/15	6-24
6.10.9	BITE Bit 16	6-24
6.11	PAYLOAD AND ORBITER TIMING BUFFERS	6-24
6.11.1	Description	6-24
6.11.2	Connectors	6-25
6.11.3	Users	6-27
6.11.4	Failure Detection	6-28
6.12	MISSION TIMERS	6-29
6.12.1	Description	6-29
6.12.2	Connector	6-30
6.13	EVENT TIMERS	6-31
6.13.1	Description	6-31
6.13.2	Connector	6-32

Section Page

6.13.3 Use 6-33
6.14 CRITICAL FAILURE MOVES 6-33
6.14.1 Both Oscillators Selected For Output 6-33
6.14.2 MTU Loss of Output 6-35
6.14.3 Erroneous MTU Output 6-36
6.15 EXTERNAL INTERFACES 6-37
6.16 REFERENCES 6-37

SECTION 6
MASTER TIMING UNIT

6.1 INTRODUCTION

Manufactured by Westinghouse, the Master Timing Unit (MTU) is the heart of the Orbiter Timing System. Not only does it provide a time source for the Orbiter, it also provides time and timing signals to various PL users. Therefore, this brief describes the various interfaces the MTU has with Orbiter and PL hardware. The discussion of software interactions shall be limited to the capabilities supported by the MTU.

6.2 DESCRIPTION

Located in Av Bay 3B, the MTU weighs 28 pounds with dimensions of 17 by 6 by 10 inches. It is dually powered by ESS 1BC and ESS 2CA and normally draws 31 watts in operate and 40 watts in warmup.

6.2.1 Operational Constraints[1]

Approximately 16 hours of warmup are required for the outputs of the quartz crystal oscillators to stabilize to an accuracy of 1 part in 10^9 per day. The operational temperature constraint of -20° to 120° F is maintained via water cooled coldplate; the MTU may be operated at 8 psia for 2 hours with a nominal operating range of 12.36 to 18 psia.

Location	AV Bay 3B
PWR	28 V 31 W
CB	ESS 1BC PNL 013:A CB 4, 3A ESS 2CA PNL 013:C CB12, 3A
Cooling	Coldplate
Weight	28 lb

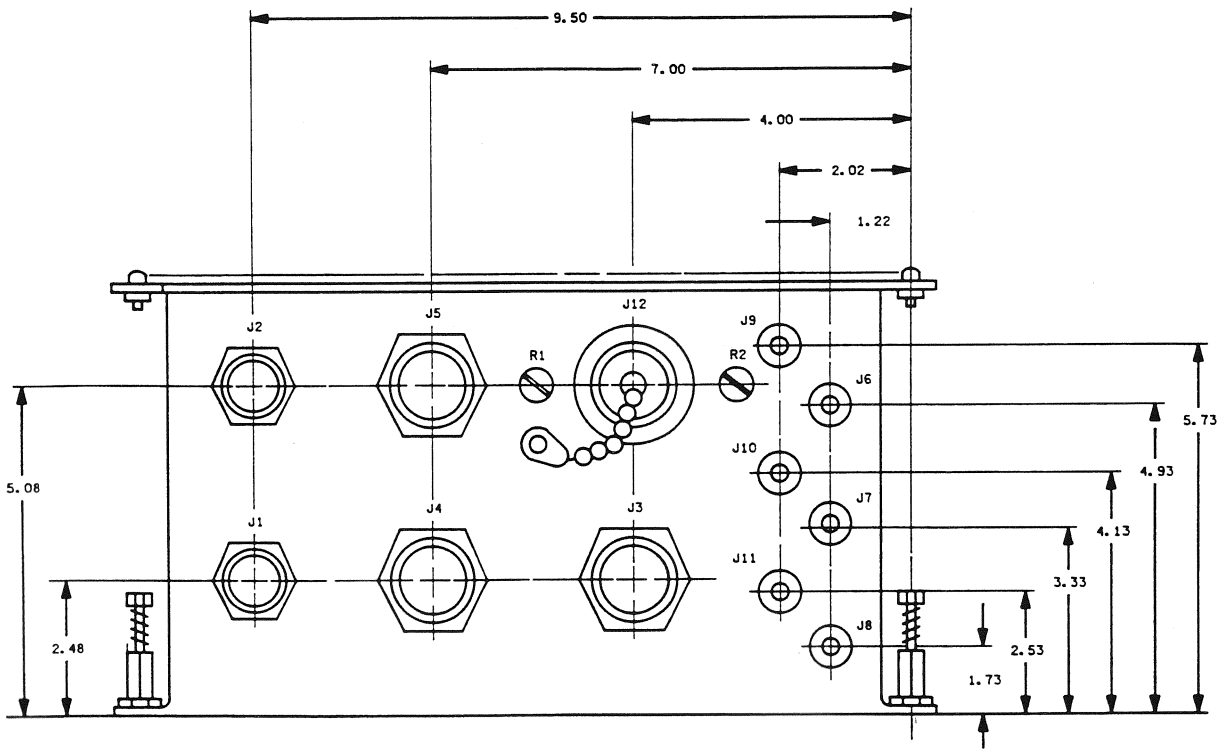


Figure 6-1.- The MTU, front view.

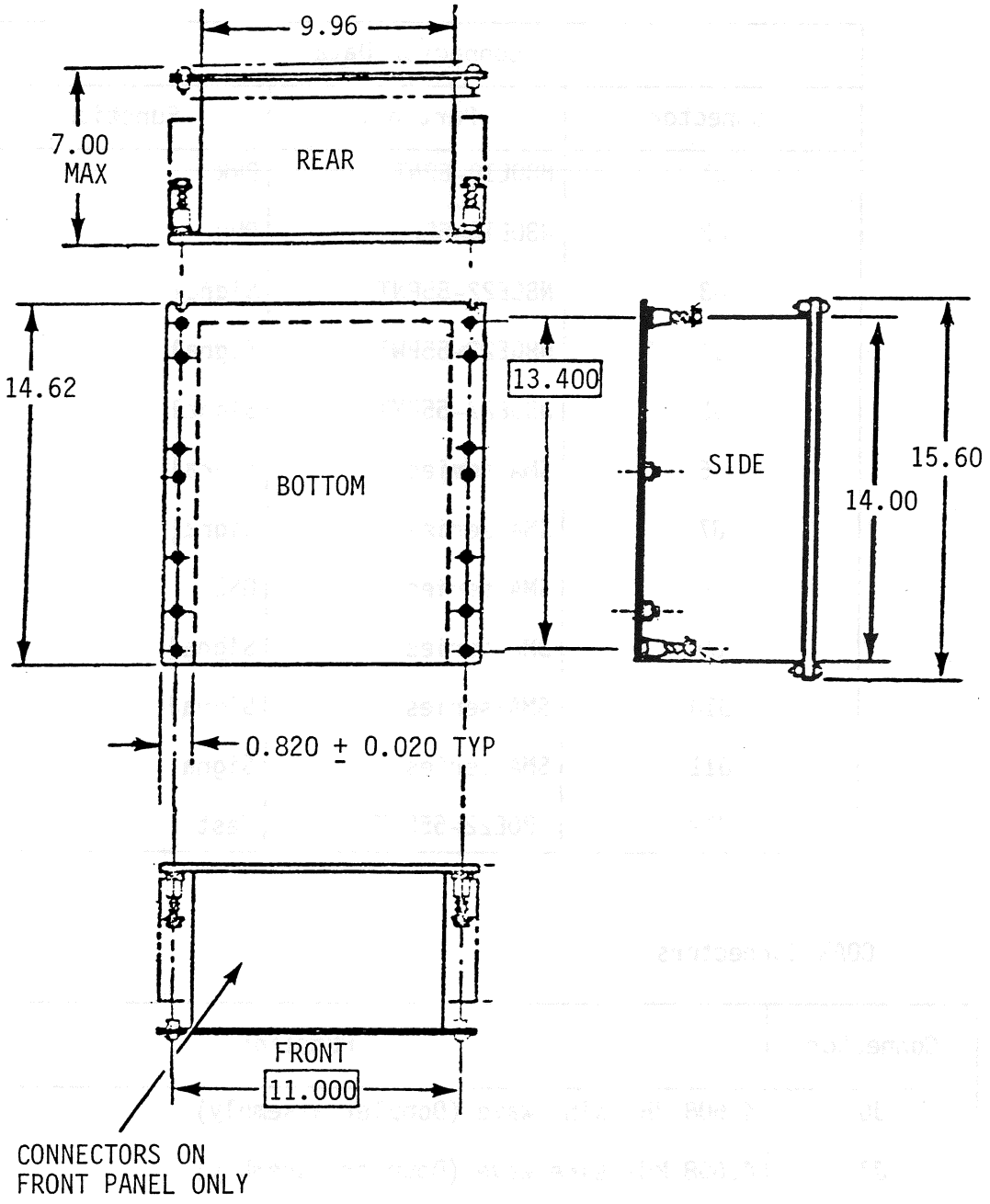


Figure 6-2.- The MTU.

6.2.2 Connectors

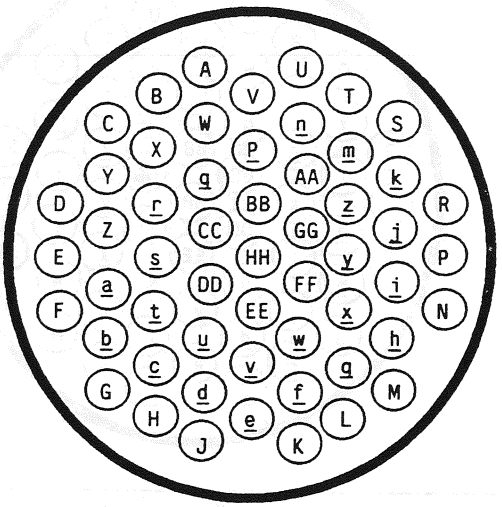
Connector Data		
Connector	Part no.	Function
J1	NBOE10-6PNT	PWR
J2	NBOE10-6PWT	PWR
J3	NBOE22-55PNT	Signal
J4	NBOE22-55PWT	Signal
J5	NBOE22-55PXT	Signal
J6	SMA series	Signal
J7	SMA series	Signal
J8	SMA series	GSE
J9	SMA series	Signal
J10	SMA series	Signal
J11	SMA series	Signal
J12	NBOE22-55PYT	Test

COAX Connectors

Connector	Function
J6	4.608 MHz sine wave (Doppler assembly)
J7	4.608 MHz sine wave (Doppler assembly)
J8	4.608 MHz sine wave (T-0 umbilical)
J9	4.608 MHz square wave (Payload Standard Distribution Panel)
J10	4.608 MHz square wave (PCMMU no. 2)
J11	4.608 MHz square wave (PCMMU no. 1)

6.2.3 Pin Definitions

A. J3 Connector - Part no.: NBOE 22-55 PNT

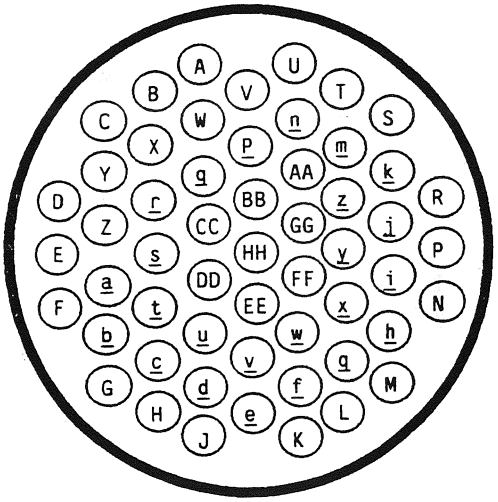


Function		Pin
FF1	BITE, GMT, MET data	A
	RTN	B
	Word discrete	W
	RTN	X
	Message in discrete	Q
	RTN	R
	Message out discrete	CC
	RTN	S
OF1 voted	BITE, GMT, MET data	U
	RTN	V
	Word discrete	N
	RTN	P
	Message in discrete	AA
	RTN	BB
OTB GMT (IRIG B)		DD
RTN		T
Network COMSEC 1 GMT (IRIG B)		U
RTN		C
OTB MET (IRIG B)		D
RTN		H

Function		Pin
Payload signal processor 1	1024 kHz	M
	RTN	G
FM signal processor	1024 kHz	H
	RTN	X
Not used	9.6 kHz	GG
	RTN	HH
Chassis ground		D
Chassis ground		E
Aft event timer	10 Hz	K
	RTN	L
BITE discrete		F
RTN		A
OSC select logic in no. 1		S
OSC seelct logic in no. 2		K
OSC select logic out		R
OSC select logic RTN		J

Figure 6-3.- J3 connector.

B. J4 Connector - Part no.: NBOE 22-55 PWT

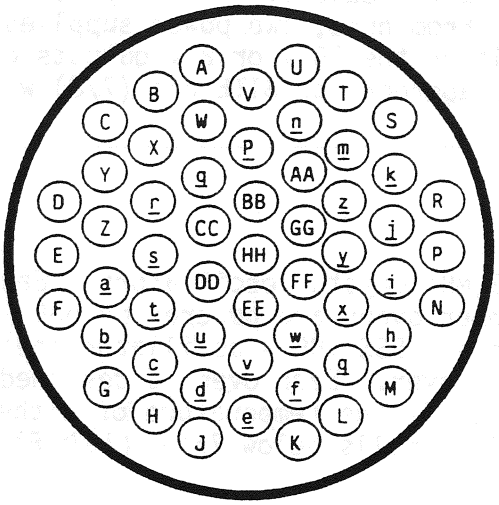


Function		Pin
FF2	BITE, GMT, MET data RTN	A B
	Word discrete RTN	W X
	Message in discrete RTN	Q R
	Message out discrete RTN	CC S
OF2 non-voted 2	BITE, GMT, MET data RTN	S T
	Word discrete RTN	K M
	Message in discrete RTN	J Z
Unused non-voted 1	BITE, GMT, MET data RTN	U V
	Word discrete RTN	N P
	Message in discrete RTN	AA BB

Function		Pin
FWD MSN timer GMT (IRIG B) RTN		DD T
PDI GMT (IRIG B) RTN		U C
FWD MSN timer MET (IRIG B) RTN		D H
Payload signal processor 2	1024 kHz RTN	M G
*	1024 kHz RTN	H X
FWD event timer	10 Hz RTN	K L
Not used	9.6 kHz RTN	GG HH
Bite discrete RTN		F A
Chassis ground		D
Chassis ground		E

Figure 6-4.- J4 connector.

C. J5 Connector - Part no.: NBOE 22-55 PXT



	Function	Pin
FF3	BITE, GMT, MET data	A
	RTN	B
	Word discrete	W
		X
	RTN	
	Message in discrete	Q
		R
	RTN	
	Message out discrete	CC
		S
	RTN	
Unused non-voted 3	BITE, GMT, MET data	U
		V
	RTN	
	Word discrete	N
		P
	RTN	
	Message in discrete	AA
		BB
	RTN	
PTB GMT (IRIG B)		U
		C
RTN		
Network COMSEC 2		DD
		T
RTN		
PTB MET (IRIG B)		D
		H
RTN		

Function		Pin
FM signal processor	1024 kHz	H
		X
	RTN	
Payload distribution panel	1024 kHz	M
		G
	RTN	
Payload distribution panel	1 kHz	FF
	RTN	EE
Payload distribution panel	100 hZ	W
		V
	RTN	
Payload distribution panel	10 Hz	K
		L
	RTN	
Payload distribution panel	1 Hz	F
		E
	RTN	
Not used	9.6 kHz	GG
		HH
	RTN	
Chassis ground		D
Chassis ground		E

Figure 6-5.- J5 connector.

6.3 POWER

Power is supplied to the MTU by ESS 1BC and ESS 2CA. This power is routed through two 3 amp circuit breakers: cb 4 on panel O13:A and cb 12 on panel O12:C, respectively. From here, two power supplies convert 28 V dc to 16 V dc and 8 V dc. If either the 16 V or 8 V outputs of the power supplies go out of tolerance, the appropriate BITE bit (7/8) will be set. (See MTU BITE Word.)

6.4 OSCILLATORS

The MTU has two 4.608 MHz crystal oscillators. Each one resides in two nested ovens. The inner ovens are powered by both ESS 1BC and ESS 2CA; the outer ovens by only one: ESS 1BC - oscillator one's outer oven, ESS 2CA - oscillator two's outer oven. Each oven is designed to maintain a temperature of 80° C (176° F). If the temperature of either member of the pair exceeds 85° C (185° F) or falls below 75° C (167° F), a BITE bit (5/6) will be set.

Each oscillator outputs a 4.608 MHz sine wave. If the amplitude of this signal falls below .8 VRMS, the BITE bit (2/3) for that oscillator will be set. Additionally, the outputs of both oscillators are compared and if found to differ by more than .04 Hz in a 60-second period, BITE bit 4 will be set. Fine tuning of each oscillator is possible during ground checkout through an external adjustment which has a resolution of ± 1 part in 10^{10}.

As either oscillator can drive the MTU, a switch (S15) on panel O6 allows the manual selection of a particular oscillator or the automatic selection of one. While this switch is in the auto position, the fault detection circuitry in the MTU will select an oscillator.

6.4.1 Select Logic

If the S15 switch is in the AUTO position, the select logic will be governed by the following truth table.

A1 = OSC 1 amplitude failed bit 2

T1 = OSC 1 temperature failed bit 5

A2 = OSC 2 amplitude failed bit 3

T2 = OSC 2 temperature failed bit 6

N/C = No change

OSC 1		OSC 2		Selected OSC		
A1	T1	A2	T2	1	2	N/C
0	0	0	0			X
1	0	0	0		X	
0	1	0	0		X	
0	0	1	0	X		
0	0	0	1	X		
1	1	0	0		X	
1	0	1	0			X
1	0	0	1		X	
0	1	0	1			X
0	0	1	1	X		
0	1	1	0	X		
1	1	1	0			X
1	1	0	1		X	
1	0	1	1			X
0	1	1	1	X		
1	1	1	1			X

It is important to note that, if temperature is failed on one oscillator and amplitude on the other, the oscillator with failed temperature will drive the accumulators.

If the switch is in the manual mode (oscillator one or two selected), the selection filter logic will be overridden. The selected oscillator's 4.608 MHz signal will be used to drive the timing outputs regardless of what failures are detected on its output.

6.5 MTU SIGNAL DISTRIBUTION

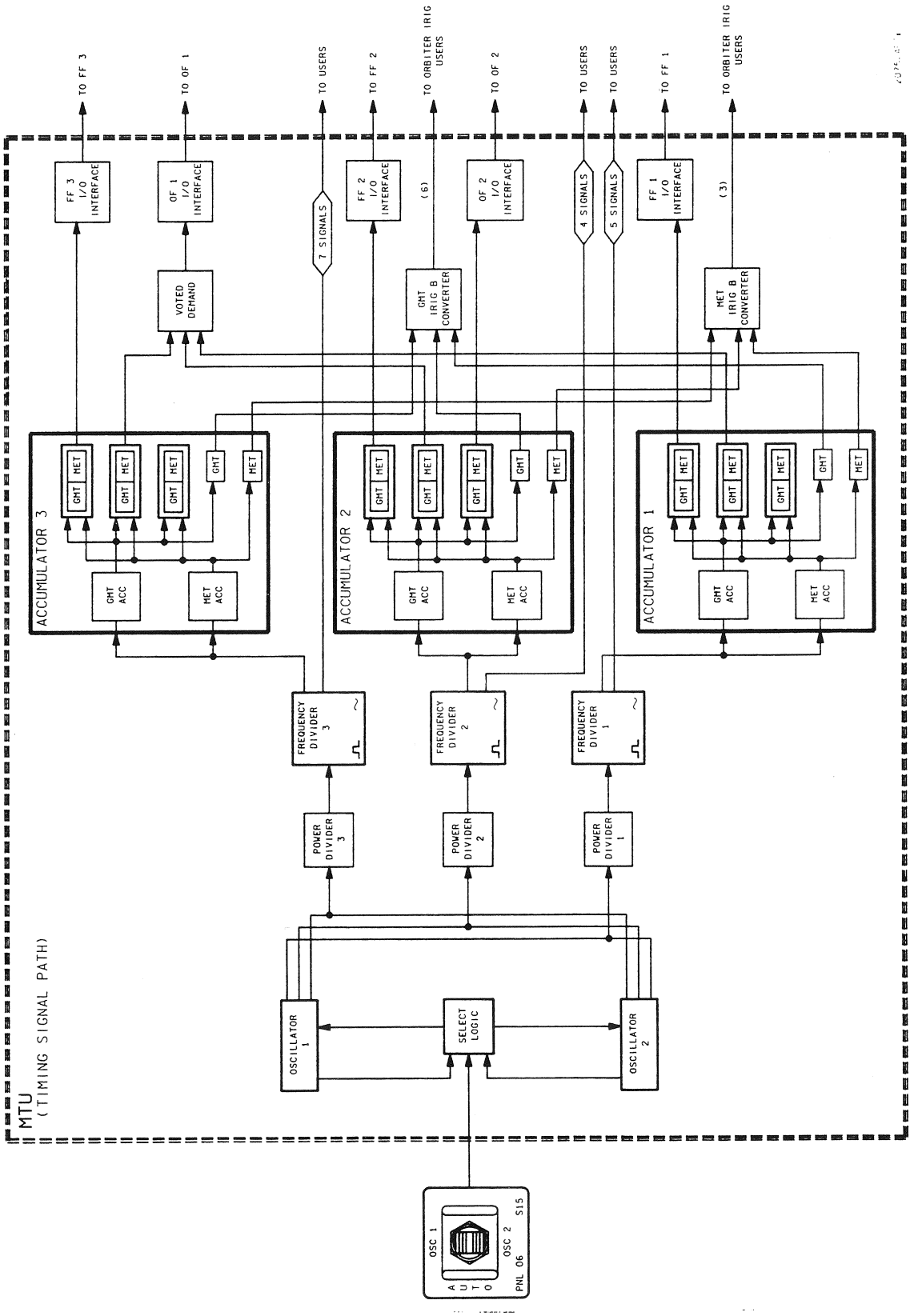


Figure 6-6.- MTU signal distribution.

6.5.1 Power Divider/Frequency Divider

The 4.608 MHz sine wave outputs of the driving oscillator are fed through three Power Dividers (PDs) which convert the signal to a 9.216 MHz square wave. Each PD supplies one of the three Frequency Dividers (FDs) (PD1/FD1, PD2/FD2, PD3/FD3).

Here, the 9.216 MHz signal is stepped down to produce 4.608 MHz, 1024 kHz, 9.6 kHz, 1 kHz, 100 Hz, 10 Hz, and 1 Hz outputs.

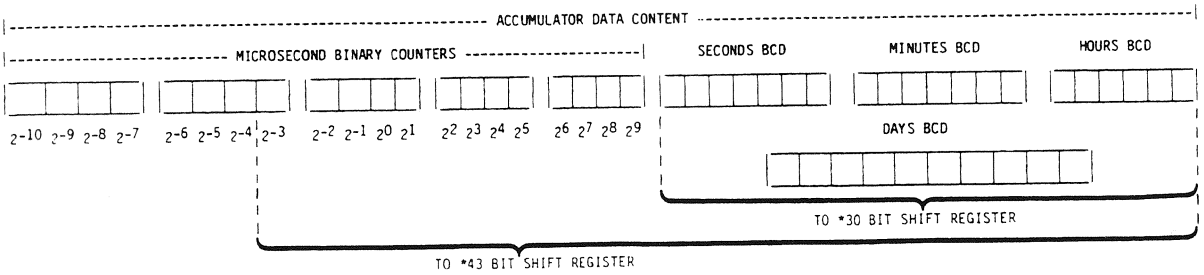
FD1	FD2	FD3
4.608 MHz - PAYLOAD 1024 kHz - PSP1 1024 kHz - FM SIGNAL PROCESSOR 1 1024 kHz - ACC1 1 Hz - DRIFT DETECTOR/ FREQUENCY COMPARE 10 Hz - AFT EVENT TIMER	4.608 MHz - PCMMU2 1024 kHz - PSP2 1024 kHz - ACC2 1 Hz - FREQUENCY COMPARE 10 Hz - FWD EVENT TIMER	4.603 MHz - PCMMU1 1024 kHz - FM SIGNAL PROCESSOR 2 1024 kHz - PAYLOAD 1024 kHz - ACC3 1 Hz - PAYLOAD/ FREQUENCY COMPARE 10 Hz - PAYLOAD

A 1 Hz signal from each FD is used by the frequency compare circuitry. In effect, the output of each FD is compared against the other two; if it disagrees, a BITE bit (13, 14, or 15) will be set to indicate the erring divider.

6.6 ACCUMULATORS

One of the 1024 kHz outputs from each FD is used by the corresponding accumulator; i.e., FD1 drives ACC1, FD2/ACC2, and FD3/ACC3. Each accumulator contains one GMT accumulator and one MET accumulator. Time is stored in these accumulators in the following form.

6.6.1 Accumulator Data Content



*DATA ARE TRANSMITTED OVER PARALLEL LINES.

Figure 6-7.- Accumulator data content.

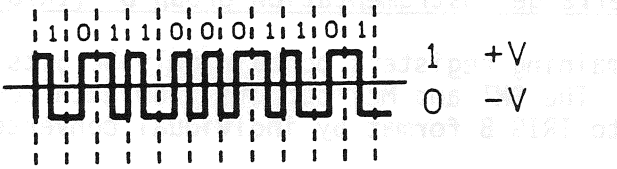
One should note that the lower registers, 2^{-10} - 2^{-4}, are not seen by any time user. At MTU powerup, these registers initialize randomly and may cause a skew in the setting of the bit for 2^{-3} (125 μsec). This is the LSB in the time words and will cause a toggling of the BSR's for ACC 1, 2, 3 fail. A one word reset affects these lower registers; an update does not.

The two accumulators transmit via parallel data lines to three shift register pairs (GMT and MET), and to one 30-bit GMT shift register and to one 30-bit MET shift register. Contained in the GMT/MET register pair are 43 bits of GMT LSB .125ms, 43 bits of MET LSB .125ms, and 16 bits of BITE.

6.7 ACCUMULATOR TIME OUTPUTS

6.7.1 FF MDMs

Register pair 1 outputs from each accumulator are transmitted serially to the dedicated FF I/O interfaces, ACC1-FF1, ACC2-FF2, and ACC3-FF3. Here the data is encoded to Manchester II for transmission to the FF MDM. For the transmittal to occur, the FF MDM must set two discretes: the "message in" discrete and the "word" discrete. When the MTU receives the "message in" discrete, the transmitter/encoder for that interface is enabled and the associated GMT/MET/BITE registers are loaded. As long as the "message in" discrete is high, the MTU may respond to data requests. Upon receipt of the "word" discrete(s), the data in the GMT/MET/BITE registers are serially sent (clocked) to the I/O interface where they are Manchester-encoded for transmission to the FF MDM. After the MDM has obtained all the requested data, the "message in" discrete will be reset (low) thereby disabling the transmitter and preventing the loading and clocking of the registers. Each MDM clocks and requests the loading of its accumulator register pair; it does not affect the others.



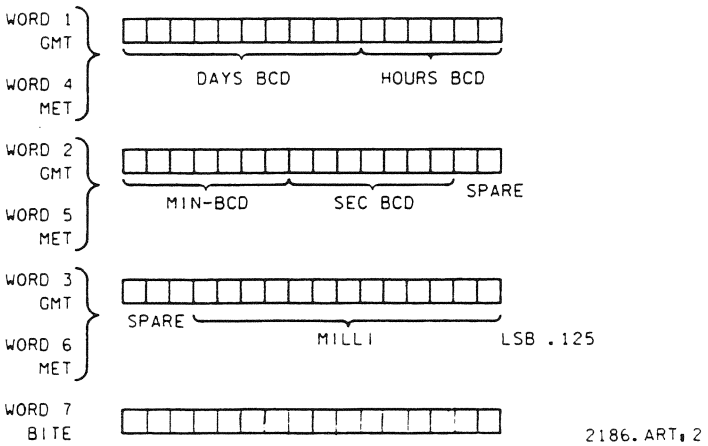
MANCHESTER II

10 = 1
01 = 0

AVERAGE TRANSMITTAL TIME
1 MICROSECOND/WORD

2185.Art 2

Data is transmitted in the following form.



2186.ART.2

6.7.2 OF1 MDM

Register pair 2 outputs are used by OI. The serial data pass through the voted demand circuitry prior to arriving at the OF1 interface. This logic checks the three ouputs on a bit by bit basis. As there are only two possible states 1 or 0, the state that at least two outputs agree on is transmitted to the OF1 interface. It should be noted that the same BITE register is used to supply all accumulator shift register pair 2's. Second, the logic detects accumulator shift register pair 2 failures, and a disagreement on any bit state will result in a BITE bit's (9,10,11) being set. Data requests from the OF1 MDM are identical to those of the FF MDM's. The only exception is: OF1 controls the resetting and loading of shift register pair 2 in all accumulators.

6.7.3 OF2 MDM

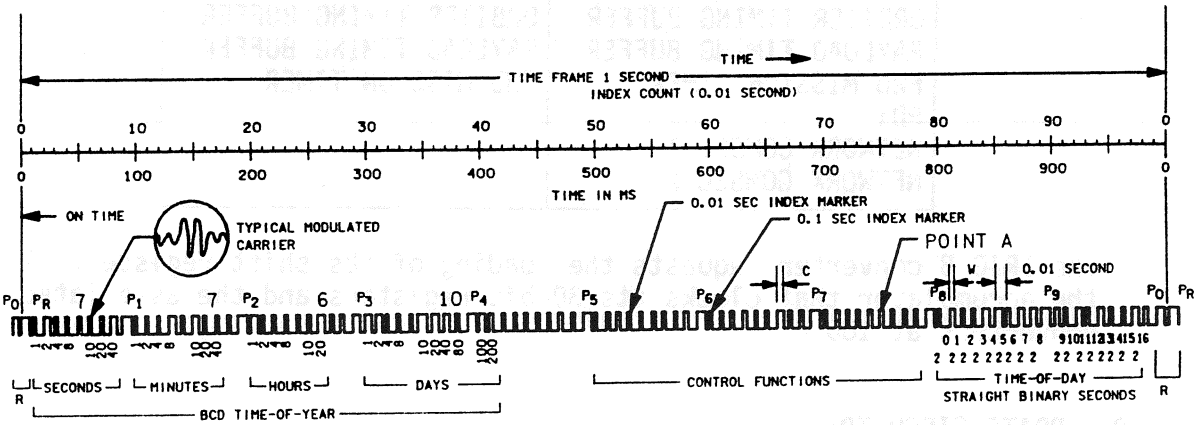
Shift register pair 3 in accumulators 1 and 3 are not used. The output of accumulator 2's register pair 3 is used by OF2. This is the OI nonvoted time. Data requests are identical to those of the FF MDMs.

6.7.4 Interrange Instrumentation Group B (IRIG B)

The two remaining registers contain only 30 bits of data each; there is no BITE word. The GMT and MET outputs from each accumulator are separately converted to IRIG B format by individual converters.

A. IRIG "B" format

The format has the following characteristics:



TIME AT POINT A

SECONDS	2 + 40
MINUTES	8 + 10
HOURS	1 + 20
DAYS	1 + 2 + 10 +20 + 40 + 100
P1-P7 (5 INDEX MARKERS)	.70 SECONDS .05 SECONDS
	173D + 21H + 18M + 42.75 SEC

2182.ART, 4

1. One pps frame reference marker (PO/PR = R)
2. BCD time of year code word (30 digits) with LSB first 1=5ms, 0=2ms
3. Control function (not used)
4. Binary time of day (17 digits) 1=5ms, 0=2ms
5. Ten pps position identifiers (P_0-P_9) 8ms
6. One hundred pps index markers 2ms

B. IRIG channel fail

The three GMT channels are compared as are the three MET channels; if any one is in error, BITE bit 12 will be set. In addition, the three GMT (MET) channels supply time to three separate output voters. The three voted IRIG B GMT signals are split to produce six signals. These, in addition to the three voted MET signals, are transmitted directly to the user.

C. IRIG B users

VOTED GMT	VOTED MET
ORBITER TIMING BUFFER PAYLOAD TIMING BUFFER FWD MISSION TIMER PDI NETWORK COMSEC 1 NETWORK COMSEC 2	ORBITER TIMING BUFFER PAYLOAD TIMING BUFFER FWD MISSION TIMER

The IRIG B converter requests the loading of its shift register. It is the accumulator that clocks its 30-bit registers and the associated converters at 100 Hz.

6.8 UPDATE CIRCUITRY

The updating of the MTU is controlled by a PASS GPC SPEC (SPEC 2 - TIME).

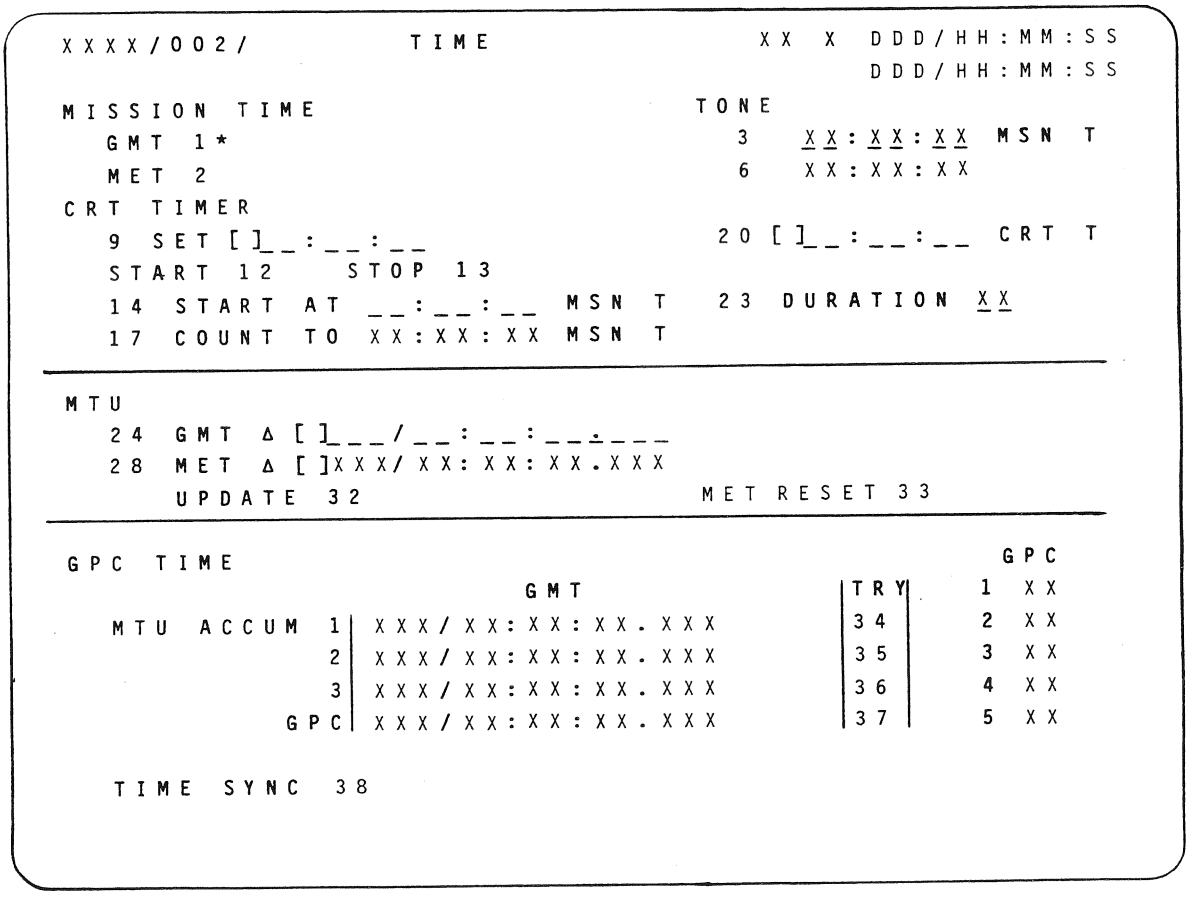
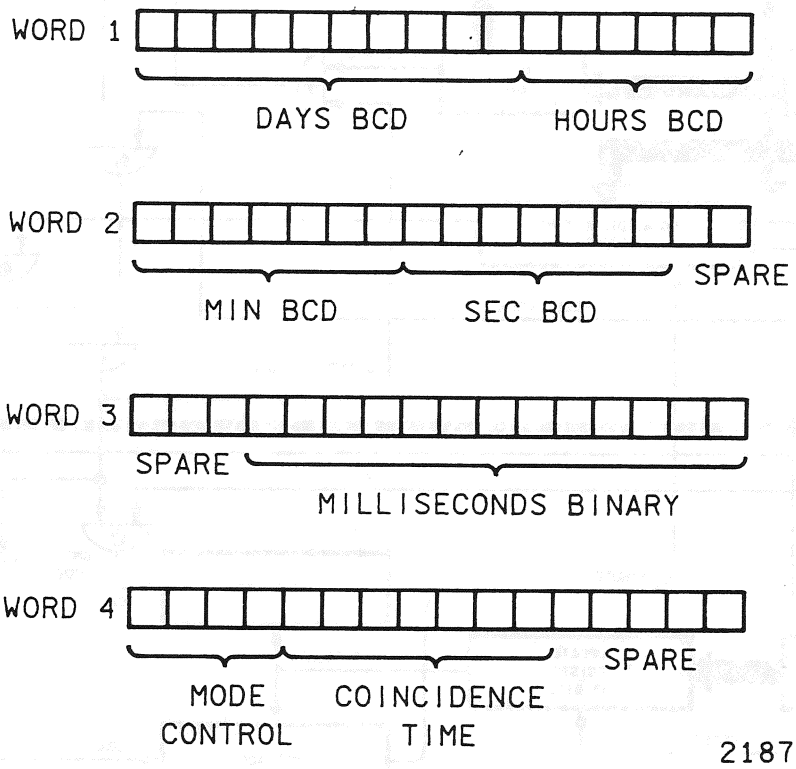


Figure 6-8.- TIME (SPEC 2) display.

The times obtained via the FF MDMs are displayed as MTU ACCUM 1,2,3. Items 24 to 33 and 38 affect the MTU.

6.8.1 Update Word Format

Word Format of Update Request



2187.ART;3

RESERVED

MODE CONTROL

0 0 1 1	GMT	UPDATE
0 1 0 1	MET	UPDATE
0 1 1 0	GMT	RESET
1 0 0 1	MET	RESET

6.8.2 Update Circuitry Overview

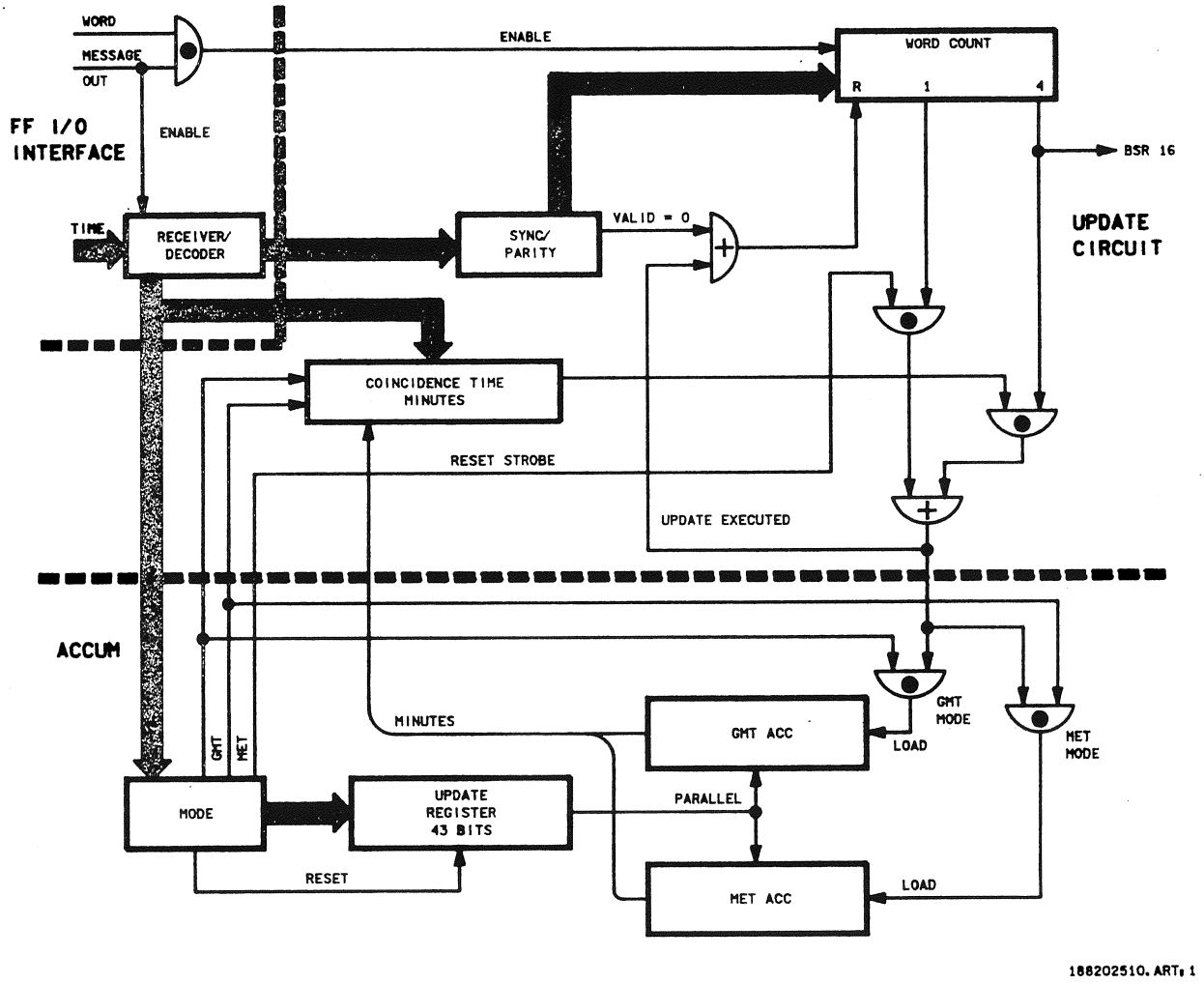


Figure 6-9.- Update circuitry overview.

6.8.3 Update Logic

For the MTU to process update, three things must occur.

A. "Word" discrete goes Hi

B. "Message out" discrete goes Hi

C. Valid control word is received

When the MTU receives the "message out" discrete (update), it enables the receiver and Manchester decoder in the appropriate I/O interface. Upon receipt of the "word" discrete, the logical AND of "word" and "message out" enables the word count logic. The MTU is now ready to validate the incoming update request. It should be noted, the lowest noncommfaulted FF MDM (1, 2, or 3) issues the update command. Once the command has been sent, both "MESSAGE OUT" and "word" discretes will be reset. Once the command has been decoded, sync and parity are checked and, if valid, the word counter reset is zeroed. Next, the number of words are checked for a 1 or a 4. If it is found to be 4, BITE bit 16 is set, indicating that an update will be performed at coincidence time. If it has a value of 1, a reset will be immediately executed. The mode control word (1 word reset or 4 word update) indicates whether the GMT or the MET accumulator will be affected. The update, total GMT or MET, is loaded into the update register (a 1 word reset will load GMT of 1 day or MET of a 0 days). Coincidence time is constantly compared against the minute value of the time to be updated. When they are equal, a load strobe will be issued, causing the time stored in the update register to be loaded via parallel lines into the appropriate accumulator. In addition, BITE bit 16 will be reset. The update logic affects only the 43-bit portion of the accumulator; the 1 word reset affects the entire accumulator. Since coincidence time is dependent on the MTU's version in comparison with the command time, an update might take as much as 60 minutes before the accumulator is actually updated.

6.9 BITE STATUS REGISTERS

There are five separate BSRs. Each is associated with a specific shift register pair.

BSR1	ACC1	Shift register pair 1
BSR2	ACC2	Shift register pair 1
BSR3	ACC3	Shift register pair 1
BSR4	ACC1,2,3	Shift register pair 2
BSR5	ACC1,2,3	Shift register pair 3

There is one important note on the use of the BITE word: there is no BITE detection on the timing outputs to the FF MDMs.

In the preceding text, a note was made indicating which signal/output had BITE performed on it.

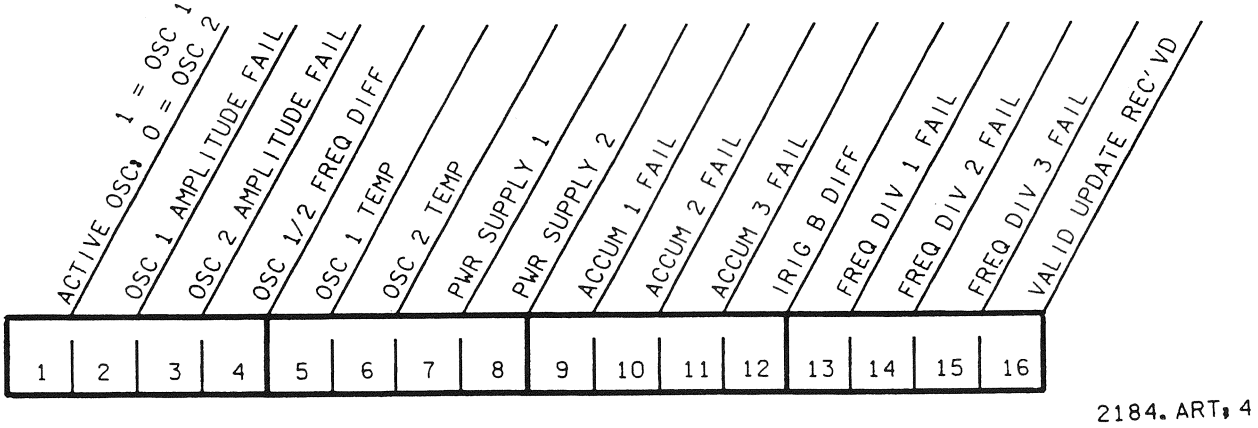


Figure 6-10.- MTU BITE word OI V75M3540P.

6.9.1 Bit Definitions

Bit	Definition
1	1 = oscillator No. 1 is used as the active driver of the MTU. 0 = oscillator No. 2 is used as the active driver of the MTU. Bits 2 through 15: '0' = Good, '1' = Failed.
2	The voltage output of OSC 1 is < 0.8V RMS. This will cause the auto select logic, if switch MASTER TIMING UNIT OSCILLATOR is in the AUTO position PNL 06, to switch to OSC 2 if OSC 1 is the driver. Bit 1 will change state to reflect the change in OSC control.
3	The voltage output of OSC 2 is < 0.8V RMS. This will cause the auto select logic, if switch MASTER TIMING UNIT OSCILLATOR is in the AUTO position PNL 06, to switch to OSC 1 if OSC 2 is the driver. Bit 1 will change state to reflect the change in OSC control.
4	The output frequency of the oscillators are compared with each other and a flag is set if they should drift apart by a relative frequency of 0.04 Hz in a 60-second period. This reflects that one of the oscillators is out of spec with a drift rate greater than 1×10^{-8}.
5	Oscillator No. 1 oven temperature has exceeded the design limits of 70° C to 85° C. This failure will cause the auto select logic, if switch MASTER TIMING UNIT OSCILLATOR is in the AUTO position PNL 06, to switch to OSC 2 if OSC 1 is the driver. Bit 1 will change state to reflect the change in OSC control.
6	Oscillator No. 2 oven temperature has exceeded the design limits of 75° C to 85° C. This failure will cause the auto select logic, if switch MASTER TIMING UNIT OSCILLATOR is in the AUTO position PNL 06, to switch to OSC 1 if OSC 2 is the driver. Bit 1 will change state to reflect the change in OSC control.
7	Power Supply No. 1 output voltage (8 or 16 volts) is out-of-limits (6.9 V to 9.2 V or 13.8 V to 18.6 V). No affect on BTU performance since power supply No. 2 is redundant.
8	Power Supply No. 2 output voltage (8 or 16 volts) is out-of-limits (6.9 V to 9.2 V or 13.8 V to 18.6 V). No affect on BTU performance since power supply No. 1 is redundant.
9	The output of accumulator No. 1 shift register pair 2 does not coincide with the outputs of accumulators 2's and 3's (bit-by-bit). This may indicate an accumulator 1 problem. If source selected by GPC is accumulator 1, then MTU RM may occur with GPC selecting another accumulator.
10	The output of accumulator No. 2 shift register pair 2 does not coincide with the outputs of accumulators 1's and 3's (bit-by-bit). This may indicate an accumulator 2 problem. If source selected by GPC is accumulator 2, then MTU RM may occur with GPC selecting another accumulator.
11	The output of accumulator No. 3 shift register pair 2 does not coincide with the outputs of accumulators 1's and 2's (bit-by-bit). This may indicate an accumulator 3 problem. If source selected by GPC is accumulator 3, then MTU RM may occur with GPC selecting another accumulator.
12	IRIG B Format generator outputs are voted bit-by-bit. If one of three channels for GMT or MET is not the same, then flag is set. This flag may coincide with bit 9, 10, or 11 being set.
13	The 1 Hz output of the frequency divider package No. 1 is compared to the 1 Hz outputs of frequency dividers No. 2 and No. 3. Both comparisons have to miscompare to set flag. It is possible to get the flag and still have no effect on the MTU outputs. When it does affect the output, accumulator No. 1 time will vary from the other accumulators causing bits 9 and 12 to set.
14	The 1 Hz output of the frequency divider package No. 2 is compared to the 1 Hz outputs of frequency dividers No. 1 and No. 3. Both comparisons have to miscompare to set flag. It is possible to get the flag and still have no effect on the MTU outputs. When it does effect the output, accumulator No. 2 time will vary from the other accumulators causing bits 10 and 12 to set.
15	The 1 Hz output of the frequency divider package No. 3 is compared to the 1 Hz output of frequency dividers No. 1 and No. 2. Both comparisons have to miscompare to set flag. It is possible to get the flag and still have no effect on the MTU outputs. When it does affect the output, accumulator No. 3 time will vary from the other accumulators causing bits 11 and 12 to set.
16	Occurs when an update other than a reset to the MTU is initiated. The flag represents a valid update (i.e., sync, parity number of words) was received by the update logic. This flag will remain 1 until coincidence time has been reached (1 to 2 minutes).

6.10 ANALYSIS OF THE BITE WORD

First, PASS GPCs do not use the BITE word. Second, the BFS uses it at initialization only. As the BITE word is not displayed to the crew, it is the responsibility of the ground to assess the impact of the BITE word. On DPS 1, the voted demand copy of the BITE word obtained through OF1 is displayed. If OF1 is lost, BSR insight into the MTU is lost. However, the FF MDM versions of the BITE register are available via the GNC variable downlist (CZ2BTMT1,2,3).

6.10.1 BITE Bit 1

BITE bit 1 is not a failure indicator. Normally, the vehicle is launched with the MTU in auto with oscillator one as the driving oscillator. There should be no concern if BITE Bit 1 is set to 0 (OSC 2) or 1 (OSC 1) and no other BITE bit is set; the function of the MTU is the same.

6.10.2 BITE Bit 2/3

If BITE bit 2(3) is set and oscillator 1(2) is the driving oscillator, an automatic switch to OSC 2(1) will occur. If OSC 2(1) was the driving OSC or if the MTU was in the manual mode, no change would be made. As these bits indicate an amplitude failure of an OSC, it does not necessarily mean the OSC has stopped outputting. If this were the case, BSR 4 would also be set. If an oscillator with failed amplitude is the driving oscillator, the MTU's time would drift with respect to true GMT.

6.10.3 BITE Bit 4

This bit indicates a difference has been detected in the 4.608 MHz outputs of the two oscillators. If BSR 13 is also set, BSR 4 cannot be trusted as a true indication of a frequency difference. The frequency difference detect logic is driven by the 1 Hz output of FD 1. Since BSR 4 does not indicate which oscillator is at fault, MTU 1 and 2's Δt's should be monitored; should a drift be detected, it would be advantageous to select the other oscillator to drive the MTU.

6.10.4 BITE Bit 5/6

BITE bit 5(6) indicates the temperature in OSC 1(2)'s nested ovens is out of limits. This may cause an auto switchover to OSC 1(2): see OSC select logic. This failure may have an effect on the OSC's output. If BSR 4 is not set, the output is good and that OSC may be used to drive the MTU. If BSR 7(8) is set, the problem is due to the outer oven losing power.

6.10.5 BITE Bit 7/8

These bits reflect a failure of either the 8 V or 16 V outputs of power supply A(B). This has no real effect on the operation of the MTU. However, if the associated CB is open, power supply A(B) may have shorted resulting in outer oven 1(2) losing power. (BSR 5(6) will be set.) KSC has experienced the associated temperature bit being set within 2 minutes of pulling the circuit breaker.

6.10.6 BITE Bit 9/10/11

These bits do not indicate that an entire accumulator (1/2/3) has failed. It reflects only a miscompare of ACC 1(2,3) shift register pair 2's output with the other two outputs. These outputs are used by the OI voted demand circuitry. MET, GMT, or BITE may be at fault. If, in addition, BSR 12 is set, the problem is either the GMT accumulator or the MET accumulator. If the PASS GPCs were using the indicated accumulator as source and a source switch occurred, then at least the GMT accumulator is failed; MET might still be good. If BSR 13(14,15) is set, then the ACC 1(2,3) failure has been caused by the failed frequency divider 1(2,3). This will affect both GMT and MET.

The Accumulator fail bits can toggle as a result of MET Reset at lift-off or a PCMMU power cycle. This toggling was seen on STS-7, STS-8, STS-26, and STS-27. The MET Reset at lift-off forces the MET shift registers to zero and also moves the PCMMU fetch window which is a maximum of 1 microsecond. On these flights, this window moved into the PCMMU BITE read acquisition time, yielding the toggling BITE phenomenon.

The miscompare in the three accumulators is due to their 1 microsecond tolerance which is present from the time the MTU is initialized. The miscompare occurs only in the 125-microsecond least significant bit of the MET shift registers in each accumulator. A maximum of two accumulators will indicate BITE and never both at one time.

PASS does not use the Accumulator fail BITE bits, and the BFS will not use an accumulator that is failed on initialization but would still be able to use the other two accumulators. A PCMMU power cycle will probably fix this problem but is not necessary.

6.10.7 BITE Bit 12

BITE bit 12 indicates a miscompare has been detected between the three IRIG B GMT converted times or between the three IRIG B MET converted times. This may be caused by a failure of any single GMT or MET 30-bit shift register. Since the IRIG B outputs are of the voted type, this bit being set does not affect the time output to the users. If BSR 9, 10, or 11 is set, the failure is at an accumulator; if BSR 13, 14, or 15 is set, the problem is at the frequency divider.

6.10.8 BITE Bit 13/14/15

BITE bit 13(14,15) will be set if the 1 Hz output of frequency divider 1(2,3) miscompares with the other two. These bits may be set with no impact on the signal sent to the accumulators. If BSR 13 is set, BSR 4 may also be set due to circuitry design.

6.10.9 BITE Bit 16

BITE bit 16 is not a failure indicator. If a four-word update command is received and validated, the bit will be set to one until the update is executed, at which time it will be reset.

The change of state from a 1 to a 0 shows that coincidence time has occurred and the update time has been strobed into the accumulator registers. It remains in this state during all periods except when an update is requested.

6.11 PAYLOAD AND ORBITER TIMING BUFFERS[4]

6.11.1 Description

These boxes simply split and amplify one IRIG B GMT and one IRIG B MET signal to produce eight IRIG B GMT outputs and four IRIG B MET outputs each. Basically, the Orbiter Timing Buffer (OTB) supplies timing to Orbiter systems (MADS, ACIP, CCTV, SEP cameras, and aft mission timer). The Payload Timing Buffer (PTB) is dedicated to payloads; its outputs are mission reconfigurable. There is no insight into the workings of the PTB. Hence, if the user detects a problem, it is his responsibility to report it to DPS.

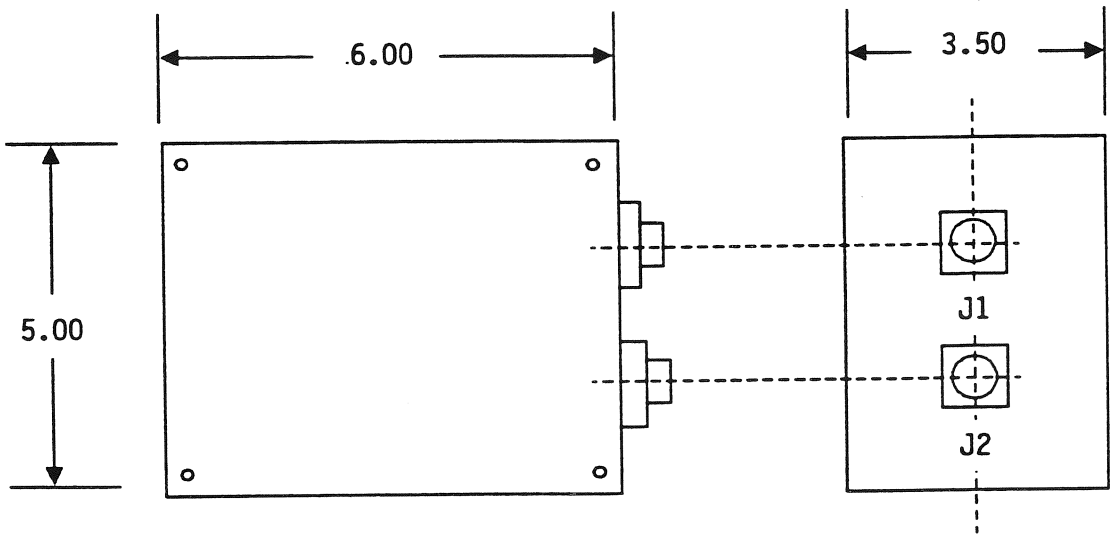


Figure 6-11.- Payload and Orbiter timing buffers.

	OTB	PTB
Location	Aft station behind PNL L16	Aft station behind PNL L16
PWR	28 V, 8 W	28 V, 8 W
CB	MNC 016 CB 27, 3A	CABPL 2, L12UR CB 3, 5 A
Cooling	Convection	Convection
Weight	5 lb	5 lb
Operational constraints	Same as MTU	Same as MTU

6.11.2 Connectors

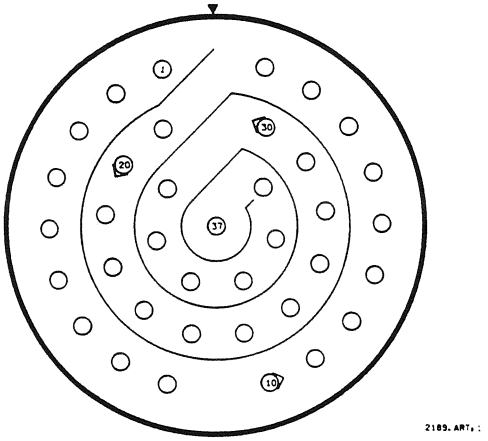
The timing buffers have two connectors:

Connector	Part no.	Function
J1	NBOE 10-6 PNT2	Power
J2	NLSOI 14-35 S	MTU I/F

A. J1 Connector- Part no.: NBOE 10-6 PNT 2

J1	
Pin	Function
A	+28 V dc
C	+28 V dc RTN
E	Case ground

B. J2 connector - Part no.: NLSOI 14-35 S



J2			
Pin	Function	Pin	Function
1	MTU GMT Hi	20	Spare
2	Lo		
		21	GMT 7 Hi
3	GMT 1 Hi	22	Lo
4	Lo		
		23	Spare
5	Signal ground		
		24	GMT 8 Hi
6	GMT 2 Hi	25	Lo
7	Lo		
		26	MTU MET Hi
8	Spare	27	Lo
9	GMT 3 Hi	28	MET 1 Hi
10	Lo	29	Lo
11	Spare	30	MET 2 Hi
		31	Lo
12	GMT 4 Hi		
13	Lo	32	Spare
14	Spare	33	MET 3 Hi
		34	Lo
15	GMT 5 Hi		
16	Lo	35	Spare
17	Spare	36	MET 4 Hi
		37	Lo
18	GMT 6 Hi		
19	Lo		

Figure 6-12.- J2 Connector.

6.11.3 Users

The outputs of the PTB are mission reconfigurable. To find the exact definition of users/pin numbers, Rockwell drawing VS72-20000X must be studied (X = flight number).

The outputs of the OTB are predefined.

User	Function	Pin
Aft mission timer	GMT 1	3 4
ACIP	GMT 2	6 7
MADS FDM	GMT 3	9 10
VSU	GMT 4	12 13
MADS PCM	GMT 5	15 16
Aft mission timer	MET 1	28 29

If a problem is detected in the output to one user, the others may be checked to verify the problem is not at the user. If all users have a problem, then either the MTU inputs to the timing buffer are bad or the buffer itself is failed. The breakers should always be checked if all timing outputs are lost.

6.11.4 Failure Detection

In the case of a payload user experiencing the problem, an in-flight maintenance (IFM) may be required. If the cb is closed and it is mandatory that time be provided to the payload, the J2 cables on the OTB and PTB may be switched. This assumes the problem is in the PTB. Once the cables have been switched, Orbiter users should be checked, for if they are getting time through the PTB, the PTB is not failed. This implies the problem is either in the cable or in one part of the PTB. If both systems are getting time, the failure is in an amplifier which is not used by Orbiter systems.

If Orbiter users are getting time but payload users are not, a pin by pin connect might be performed. This entails connecting the MTU GMT/MET from the Orbiter timing buffer cable to the PTB (provides "good" MTU signal). The pins for the user would then be connected to an appropriate set of outputs on the PTB cable (provides "new" circuitry for signal). If this does not work, the failure is associated with the payload user or the line to the payload user; the output cannot be recovered in this case.

Neither of these IFMs have been preflight approved.

6.12 MISSION TIMERS[3]

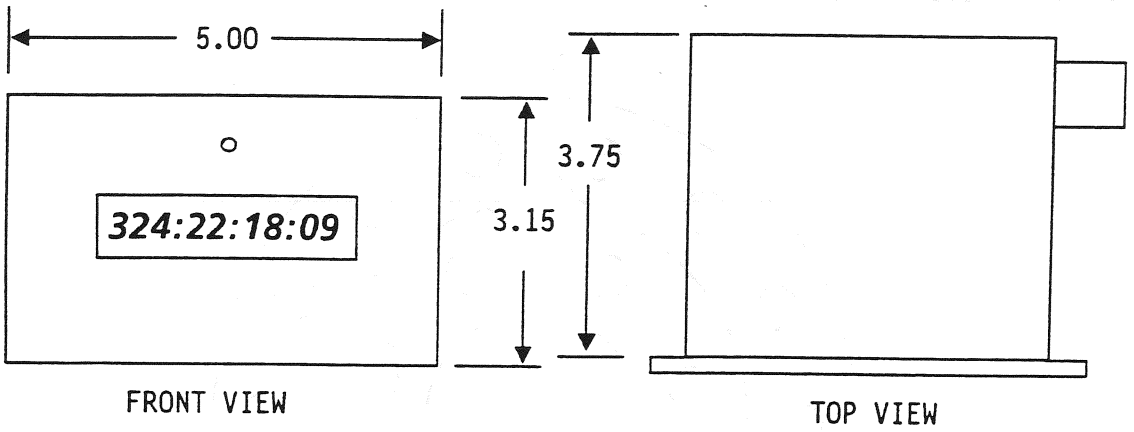


Figure 6-13.- Mission timer.

6.12.1 Description

There are two mission timers, one forward and one aft. The forward mission timer is driven by a dedicated MTU GMT/MET IRIG B signal; the aft mission timer receives its signals via the OTB.

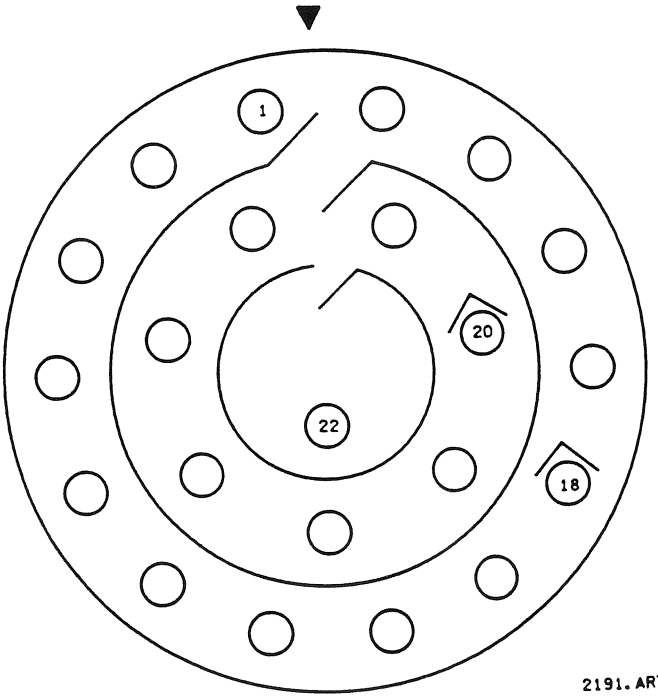
	FWD	AFT
Location	PNL O3	PNL A4
PWR	4 W 28 V dc 6.5 W 5 V ac	4 W 28 V dc 6.5 W 5 V ac
CB	MNA O14B CB12 3 A	MNB O15B CB12 3 A
Weight	2.3 lb	2.3 lb
Operational constraints	Same as MTU	Same as MTU

When selected to GMT/MET, the mission timer checks the appropriate IRIG B signal for valid format. If it is found to be invalid, the red light on the faceplate will illuminate. This light will also be lit if a failure is found in a lamp driver. If a lamp test is performed, all segments in the drivers will illuminate, including the red error light.

It is important to remember that lamp power is provided by AC1φB for the forward timer and by AC2φC for the aft timer. Even though lamp power is lost, the electronics will count as long as the IRIG B signal is received. If the IRIG B signal is lost, the mission timer will show static time; the internal oscillator is used to validate the incoming timing signal only.

6.12.2 Connector

Part no.: NLSOT 1235 P



Pin	Function	Pin	Function
1	Case ground	12	GMT
		13	RTN
2	+28 V dc		
3	RTN	14	MET
4	5 V ac, 400 Hz	15	Spare
5	RTN	16	Spare
		17	Spare
6	Spare		
		18	GMT switch
7	Control voltage		
		19	Data out test
8	MET switch		point
9	Lamp test switch	20	Spare
10	Spare	21	MET RTN
11	Spare	22	Spare

Figure 6-14

6.13 EVENT TIMERS[2]

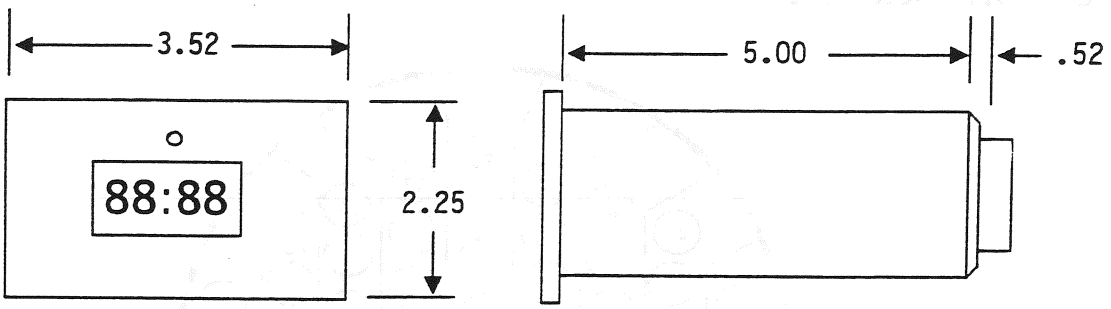


Figure 6-15.- Event timers.

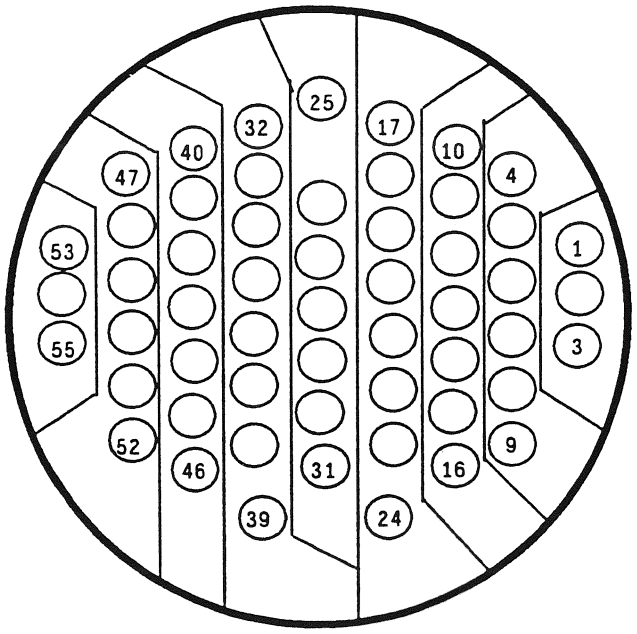
6.13.1 Description

As in the case of the mission timers, there are two event timers, one forward, one aft. These timers do not display time; rather they count down to an event or up from an event. Each one receives a 10 Hz square wave from the MTU. When this signal is found to be invalid, the red error light will illuminate. If this signal is lost, the event timer's internal oscillator will drive the count up/down logic.

	FWD	AFT
Location	PNL F7	PNL A4
PWR	4 W 28 V dc 3 W 5 V ac	4 W 28 V dc 3 W 5 V ac
CB	MNB 015:B CB 13 3 A	MNB 014:B CB 13 3 A
Weight	1.7 lb	1.7 lb
Operational constraints	Same as MTU	Same as MTU

6.13.2 Connector

Part no.: NLSOT 1635 P



Pin	Function	Pin	Function
1	Case ground	29	Thumbwheel switch, 4 minutes
2	Plus 28 V dc	30	Thumbwheel switch, 8 minutes
3	Dc return	31	Thumbwheel switch, 10 minutes
4	5 V ac 400 Hz	32	Lift-off signal
5	Ac return	33	Lift-off return
6	Spare	34	Spare
7	Spare	35	Lamp test switch
8	Spare	36	Set switch
9	Control voltage return	37	Reset switch
10	Spare	38	Thumbwheel switch, 20 minutes
11	Spare	39	Thumbwheel switch, 40 minutes
12	Spare	40	Abort signal
13	Thumbwheel switch, 1 second	41	Abort return
14	Thumbwheel switch, 2 seconds	42	Spare
15	Thumbwheel switch, 4 seconds	43	Up switch
16	Thumbwheel switch control voltage	44	Down switch
17	MTU signal	45	Start switch
18	MTU return	46	Stop switch
19	Spare	47	Spare
20	Thumbwheel switch, 8 seconds	48	Spare
21	Thumbwheel switch, 10 seconds	49	Spare
22	Thumbwheel switch, 20 seconds	50	Spare
23	Thumbwheel switch, 40 seconds	51	Spare
24	Thumbwheel switch, 1 minute	52	Control voltage
25	Spare	53	Count test point
26	Spare	54	Load thumbwheel switch data test point
27	Spare	55	Countdown equals zero test pint
28	Thumbwheel switch, 2 minutes		

Figure 6-16

6.13.3 Use

To use the timers, a time must be entered via the thumbwheels (max 59:59). When the timer switch is placed in the SET position, this input time is loaded into the timer. Next, the mode switch must be selected to either up or down. The test position is a bulb test which causes the red error light and all driver segments to illuminate. Additionally, if a failure is detected in any lamp driver, the red light shall be illuminated. When the control switch is placed to start, the timer will begin to count. The timer is stopped when the control switch is placed to stop; a timer reset will cause 00:00 to be loaded and the timer to stop counting.

The forward mission timer has two override functions: ABORT RESET and LIFT-OFF START. Presently, only the LIFT-OFF START is supported by software. When the ABORT RESET is implemented, it will take precedence over the lift-off start discrete. In either case, the timer resets and begins to count up from zero.

As in the case of the mission timers, the event timer display electronics is powered by AC1ΦB for the forward timer and by AC2ΦC for the aft timer.

6.14 CRITICAL FAILURE MODES

Nominal failure modes of the MTU are covered in detail in the DPS System Console Procedures. This section will discuss certain critical failure modes that are known to exist. The figures in this section come from Space Shuttle Systems Handbook Drawing 8.9.

6.14.1 Both Oscillators Selected For Output

This failure mode can result from two different points in the MTU: the oscillator select switch and the select logic.

It is possible for the MTU oscillator select switch on PNL O6 to fail such that both MTU Oscillators are selected for output. This failure is illustrated in figure 6-17. In this case, the switch has failed so that power from the voltage regulator is enabled to both switch input lines on the select logic. The result is that the select logic selects both oscillators for output. An IFM to the Oscillator Select Switch is not likely because the switch is not accessible behind PNL O6.

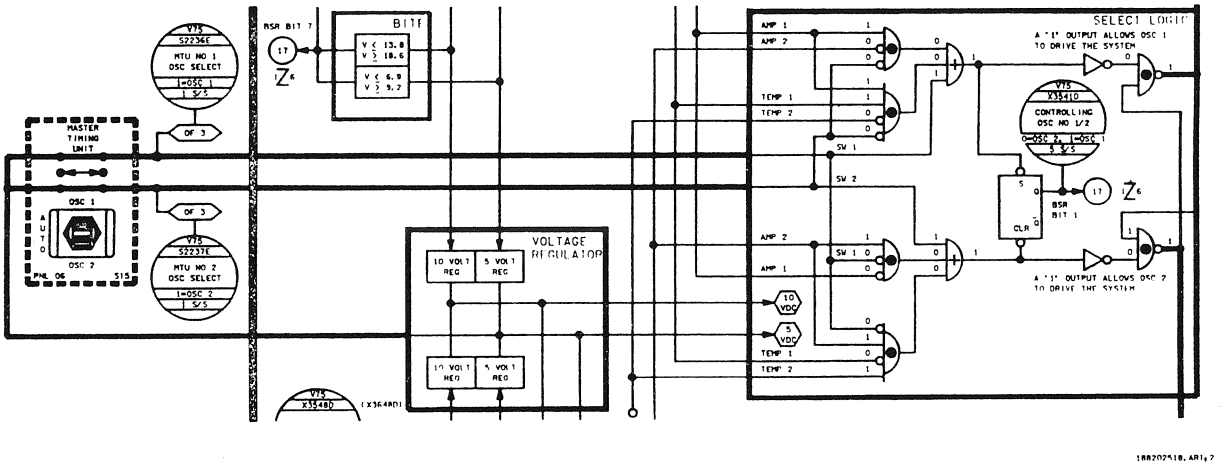


Figure 6-17.- MTU oscillator select switch failure.

The other failure that can result in both oscillators selected for output occurs in the select logic itself. The failure is illustrated in figure 6-18 and would have to be a break in the line just prior to the last logical gate for the oscillator that is not selected for output. In this example, oscillator 1 is selected for output and the line breaks before the gate for oscillator 2. This would result in a change in state out of the gate from a 0 to a 1 and oscillator 2 would be selected for output also. There is no IFM to fix this problem.

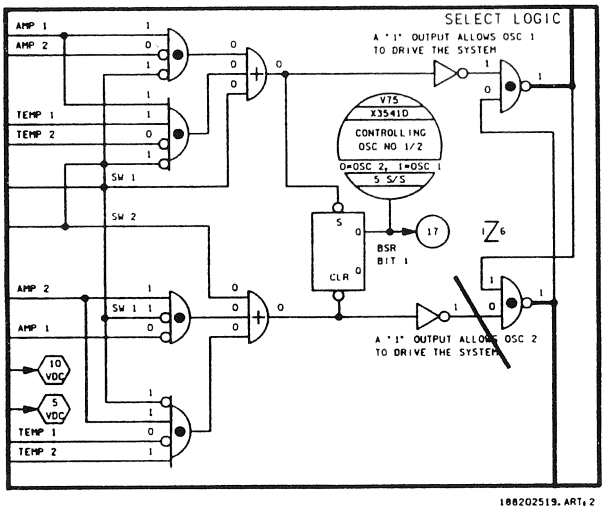


Figure 6-18.- MTU select logic failure.

In both of these failure cases, there are not any problems as long as both oscillators remain in sync with each other. If the oscillators go out of sync, the timing outputs to the users would not be usable. The only option is to totally remove power from the MTU by popping circuit breakers MTU A and MTU B on PNL O13, row A and C. This will result in no output from the MTU which is discussed in section 6.14.2

6.14.2 MTU Loss Of Output

There are two single point failures in the main MTU voltage regulator that can result in a total loss of power to the MTU. This voltage regulator is illustrated in figure 6-19. The voltage regulator is not in sufficient detail to illustrate the exact failure.

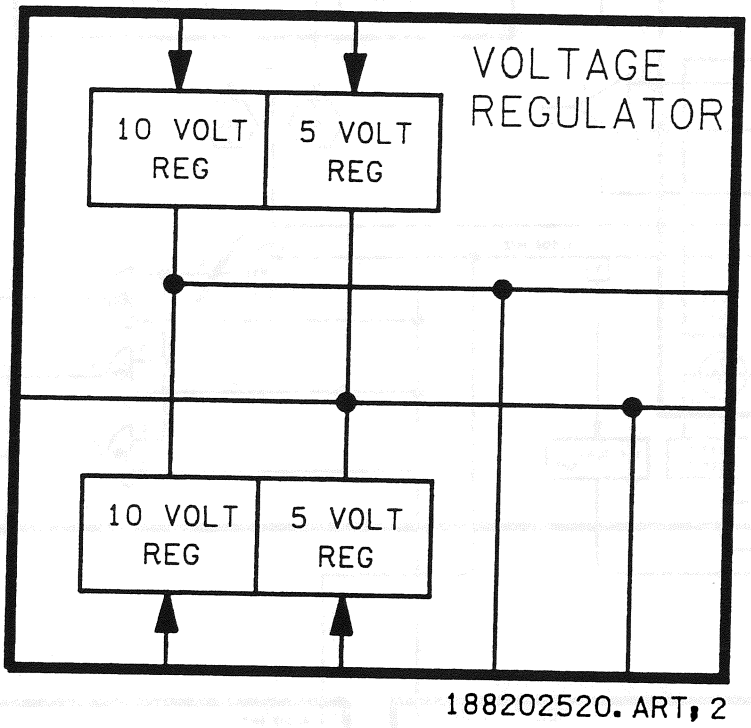


Figure 6-19.- MTU voltage regulator.

The result of this failure is that the MTU stops outputting and looks exactly like it would if you removed power to the MTU. Effects of MTU loss of output is discussed in section 6-15.

Another possible failure is in the oscillator logic gates as illustrated in figure 6-20. This failure occurs downstream of the pick off for the select logic and the drift detector and results in loss of output from the selected oscillator and no auto switch over. The system can be forced over to the other oscillator by using the MTU oscillator select switch on PNL 06.

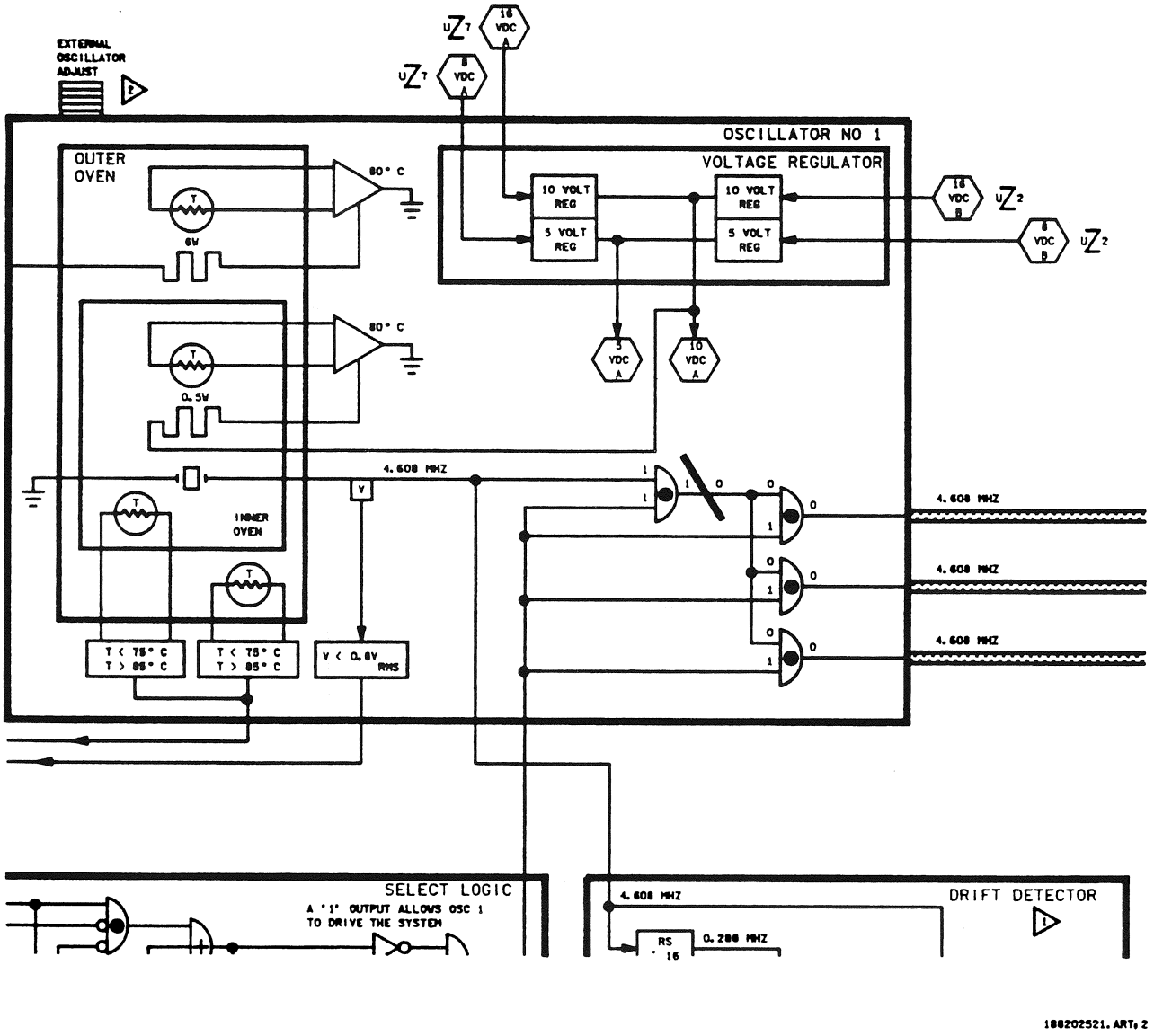


Figure 6-20.- MTU oscillator logic failure.

6.14.3 Erroneous MTU Output

There is the possibility of a change in the selected oscillator frequency output downstream of the select logic and the drift detector. This failure would have to be detected by DPS on the ground by comparing the onboard time to the ground time. This failure can be corrected by manually selecting the oscillator or by using the MTU oscillator select switch on PNL 06.

6.15 EXTERNAL INTERFACES

The MTU provides GMT, MET, and frequency to many different users on the Orbiter. Figure 6-21 illustrates the direct interfaces between the MTU and other users.

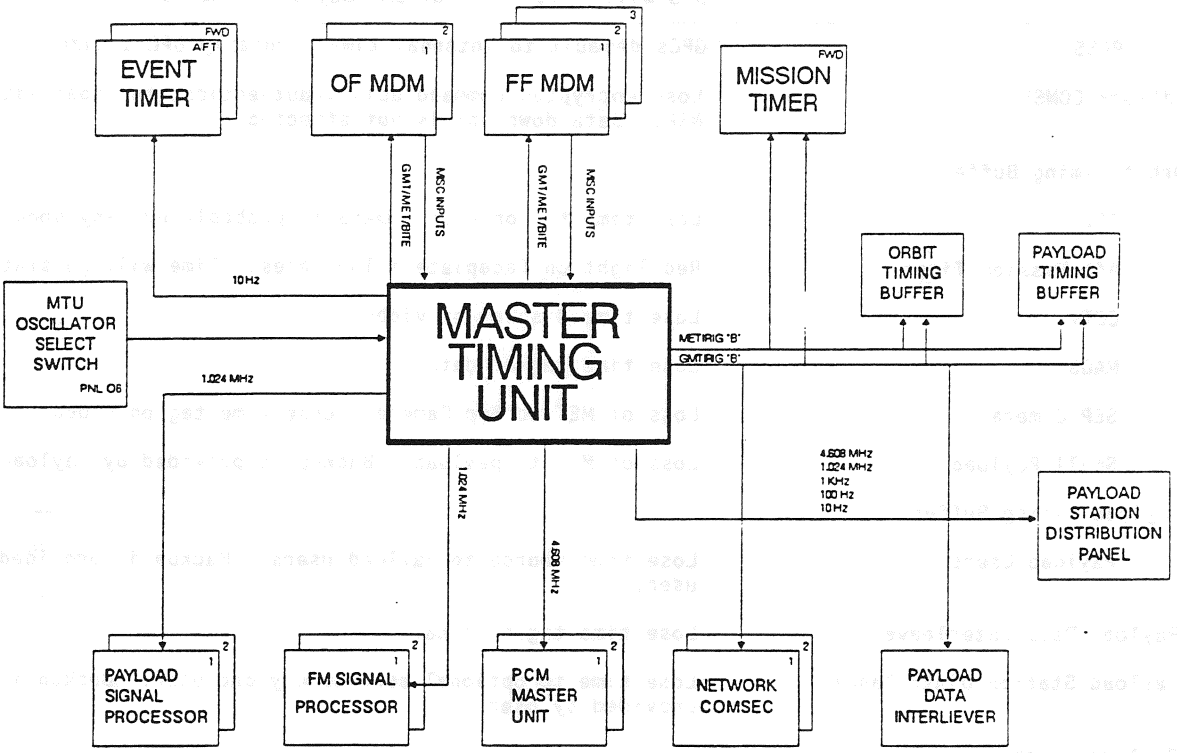


Figure 6-21.- MTU interface block diagram.

Table 6-I lists MTU loss of output effects to its users. This list is not all inclusive because of flight specific users that may be using outputs from the MTU. This information was derived from a Rockwell International report titled MTU Failure Effect Analysis dated February 16, 1987.

6.16 MTU References

1. Master Timing Unit Specification, MC454-0051.

2. Event Timer Specification, MC456-0053.

3. Mission Timer Specification, MC456-0054.

4. Buffer-Payload Timing Specification, MC456-0060.

TABLE 6-I.- MTU USERS AND LOSS OF OUTPUT EFFECTS

User	Effect
GPC's	
BFS	Default to BFS internal time. If loss of output occurs on orbit while BFS is in Halt, BFS will be no go for engage because of loss of guidance. BFS will initialize at GMT Day 1 and MET 0.
PASS	GPCs default to internal time (probably GPC 1 time).
Network COMSEC 1/2	Lose encrypted command uplink/authentication capability to NSP. Data downlink is not affected.
Orbit Timing Buffer	
ACIP	Lose time tag on data. Data is probably not any good.
Aft Mission Timer	Red light on faceplate illuminates. Time will go static.
CCTV	Lose time display on video.
MADS	Lose time tag of data.
SEP Camera	Loss of MET to Sep Camera. Lose time tag on video.
Small Payload	Loss of MET to payload. Backup is provided by payload.
Payload Timing Buffer	
Payload Users	Lose time source to payload users. Backup is provided by user.
Payload Data Interleaver	Lose time tag on data.
Payload Station Distr Panel	Lose time to optional service payload user. Backup is provided by user.
Payload Users	
IUS	Internal clock and time updates via state vector exchange. Loss of TDRS backup link during predeployment.
OSTA-3	Loss of GMT input to pallet timing buffer.
PAM	Lose data time tag to payload recorder. Time tag via GPC downlist, PCMMU, and ground station.
Spacelab	SSC an EC defaults to internal time but maintain sync through PCMMU BSR read.
SPARX-1	Loss of time tag of experiment data to MOMS recorder. Backup is telemetry clocks and ground. Loss of command to CDU unit MDMS experiment.
SPAS	Loss of time tag of experiment data. Backup is internal clock, PDI, and ground clock. Loss of command to SPAS CDU unit, recorders, and MAUS unit.
PCMMU 1/2	Defaults to internal time. Pseudo sync is lost. Downlist I/O frame starts slipping with respect to PCM master frame at the prevailing PCM-GPC oscillator drift rate.

TABLE 6-I.- Concluded

User	Effect
Signal Processors	
FM Signal Processor 1/2	Lose 1024 kHz subcarrier and lose TLM of real time data from main engine 3. Data is recorded on OPS recorder.
Payload Signal Processor 1/2	Lose command link but not return to attached and detached payloads.
Timers	
Aft Event Timer	Red error light on faceplate illuminates. Internal oscillator will drive logic.
Forward Event Timer	Red error light on faceplate illuminates. Internal oscillator will drive logic.
Forward Mission Timer	Red light on faceplate illuminates. Time will go static.

7

MULTIPLEXER/ DEMULTIPLEXER

SECTION 7
MULTIPLEXER/DEMULTIPLEXER

CONTENTS

Section		Page
7.1	OVERVIEW	7-1
7.1.1	Function	7-1
7.1.2	System	7-4
7.2	MIA	7-6
7.2.1	Receive Mode	7-9
7.2.2	Transmit Mode	7-9
7.3	SCU	7-10
7.3.1	Microprogram Control	7-10
7.3.2	MIA Interface Logic	7-11
7.3.3	Internal Bus Data Flow Control	7-12
7.3.4	Out Bus/In Bus Control	7-13
7.3.5	IOM Interface	7-14
7.3.6	Cyclic Compare Logic	7-14
7.3.7	Power Interrupt/Monitor Logic	7-14
7.3.8	Systems Clock	7-15
7.4	A/D CONVERTER	7-15
7.5	INPUT/OUTPUT MODULES	7-15
7.5.1	Discrete Input Module	7-15
7.5.2	Discrete Output Module	7-15
7.5.3	Analog Input Modules	7-16
7.5.4	Analog Output Differential (AOD) Module	7-16
7.5.5	Serial Input/Output (SIO) Module	7-16
7.5.6	Tacan/Radar Altimeter (TAC) Module	7-16
7.5.7	Input/Output Module Configuration	7-17
7.6	POWER SUPPLY	7-17
7.6.1	Core Power Supply	7-19
7.6.2	IOM Power Supply	7-20
7.6.3	SRB MDM Power Separation	7-21
7.6.4	Power-On Sequence	7-22
7.7	BUILT-IN TEST EQUIPMENT (BITE)	7-22
7.7.1	BITE Overview	7-22
7.7.2	BSR File	7-23
7.7.3	SCU BITE	7-25
7.7.4	Core Power Supply BITE	7-27
7.7.5	A/D Converter BITE	7-27
7.7.6	IOM BITE	7-27
7.7.7	SCU PROM BITE	7-28
7.7.8	BSR BITE	7-28
7.8	SEV BITS	7-28
7.9	EMDM ENHANCEMENTS	7-28
7.9.1	Gate Array Logic	7-28
7.9.2	IOM Power Supplies	7-29
7.9.3	Additional Circuit Designs	7-29
7.9.3.1	SCU Word Wrap	7-29

Section		Page
7.9.3.2	IOM Mapping	7-29
7.9.3.3	Channel Mapping	7-33
7.10	PHYSICAL CHARACTERISTICS	7-33
7.10.1	Connectors	7-36
7.10.1.1	Power Connectors	7-36
7.10.1.2	Data Bus Connectors	7-36
7.10.1.3	Signal Connectors	7-38
7.10.2	Controls	7-39
7.10.3	Cooling and Locations	7-41
7.11	FAILURE MODES	7-48
7.11.1	A/D Converter Failure	7-48
7.11.2	SRB MDM Power Supply Failure	7-52
7.11.3	MDM Power Cycling	7-52
7.11.4	EMDM Serial Input/Output (SIO) Power Supply Failure	7-55
7.11.5	EMDM S/N 221 SIO Card 11	7-55

SECTION 7
MULTIPLEXER/DEMULTIPLEXER

7.1 OVERVIEW

7.1.1 Function

The MDM is a data bus terminal unit (BTU) that performs data acquisition, data distribution, and signal conditioning functions on demand. These functions are performed under the direction of a controlling unit; i.e., an input/output processor (IOP), pulse-code modulation master unit (PCMMU), or the ground support equipment (GSE). The MDM provides bilevel discrete, analog, and serial digital inputs to and outputs from the controlling unit to the orbiter subsystems. A shielded pair of wires, called a data bus, is the only interface between an MDM and its controlling unit. The data bus supports two-way communications which must always be initiated by a controlling unit.

The MDM is a digital, dedicated processor; i.e., a special purpose digital computer designed to perform three basic operations; multiplexing, demultiplexing, and data buffering/format conversions of data received from and commands sent to the various space shuttle subsystems. The MDM design also accommodates a variety of special purpose tests known as BITE (built-in test equipment) functions.

Multiplexing occurs when the MDM's controlling unit requests analog or discrete information from a subsystem. The information is packed into a serial digital input response data word that is returned to the controlling unit on the serial data bus. Demultiplexing takes place when serial digital information sent by the MDM's controlling unit is unpacked by the MDM and output to a specific avionics system. This information from the controlling unit is referred to as an output command data word and, when unpacked, will be in the form of analog, discrete, or serial digital data. The data buffering and format conversion function is provided by the MDM for serial digital input and output channelized information.

Space shuttle MDM's currently consist of three basic types: Orbiter MDM's (DPS MDM's, which support avionics data acquisition and commanding, and OI MDM's, which support orbiter telemetry acquisition, are classified as orbiter MDM's); solid rocket booster (SRB) MDM's (SRB MDM's provide limited, specific command interfaces and SRB telemetry acquisition during MM 101 and MM 102. Note that most SRB avionics command and data acquisition is routed through the orbiter FA MDM's, not the SRB MDM's.); and flexible MDM's -- more commonly referred to as FLEX MDM's or FMDM's (Flex MDM's technically are not part of an NSTS element, but rather, a payload interface and are furnished by and at the expense of the payload customer).

It should be noted that, due to parts obsolescence, the current MDM's caannot be remanufactured. Thus for OV-105, a new generation of orbiter MDM's, known as the enhanced multiplexer/demultiplexer (EMDM), was

developed. These new MDM's are "functionally equivalent" to the current MDM's and utilize state of the art technology to enhance their reliability. Also, the EMDM's contain additional circuitry to minimize existing internal design failures associated with the current MDM's.

The only vehicle that has an entire set of EMDM's is OV-105, the Endeavor. The MDM's will be maintained on the other orbiters and replaced with EMDM's when the need arises. Therefore, in the near future OV-102, OV-103, and OV-104 may contain both MDM's and EMDM's.

This systems brief will cover the MDM and the EMDM concurrently, since they are "functionally equivalent". The term "MDM" will be used generically in this brief. However, if there is a difference between the MDM and the EMDM, it will be noted as "current MDM" and "EMDM."

The current orbiter MDM is made up of 26 hybrid circuit modules, which are packaged by function on printed circuit cards (fig. 7-1). There are 16 input/output modules (IOM's), two multiplexer interface adapter (MIA) modules, two sequence control unit (SCU) modules, two analog-to-digital (A/D) converter modules, two core power supply modules, and two modules that contain the power supplies for the IOM's.

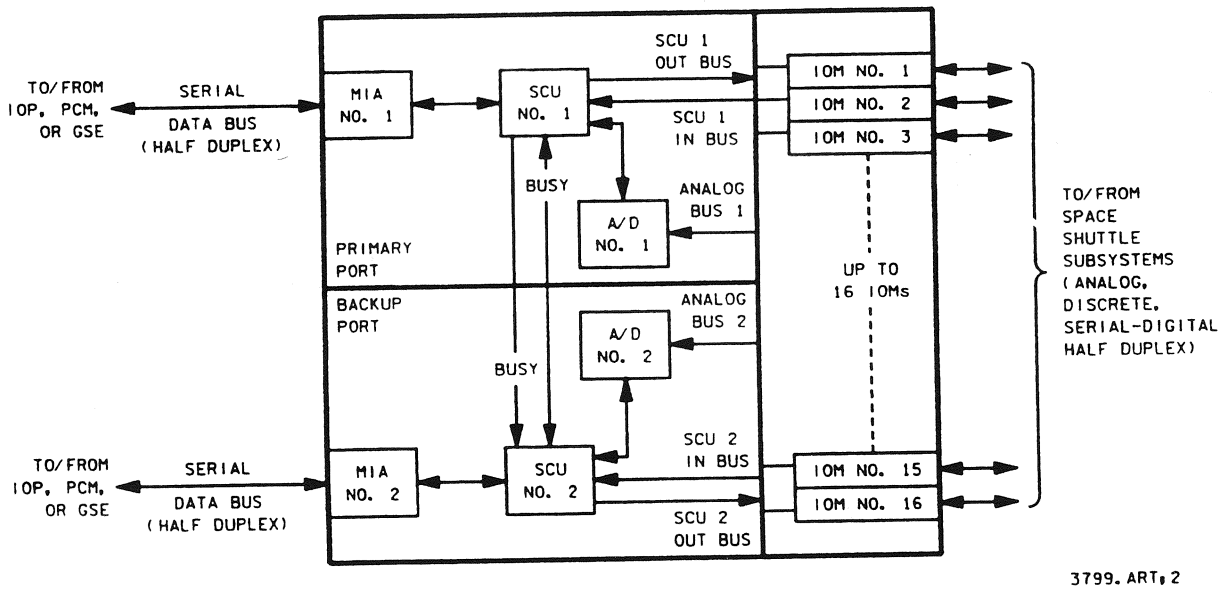


Figure 7-1.- MDM basic design.

However, the orbiter EMDM is made of only 24 circuit modules. These modules consist of both hybrid and gate array logic. Many of the digital hybrids in the current MDM were replaced with gate array logic to enhance long-term producibility and reduce the parts count. The EMDM has the same modules as the MDM with the exception of the two modules that contain the power supplies for the IOM's. In the EMDM, these IOM power supplies are mounted directly on the IOM cards.

The SRB MDM is different from an orbiter MDM because it has only 8 IOM's instead of 16. The SRB MDM has a total of 18 hybrid circuit modules.

The flexible MDM, known as a FLEX MDM, consists of 12 modules; 8 IOM's, 1 MIA, 1 SCU, 1 A/D converter, and 1 core power supply module. The power supplies for the IOM's are physically located on each module.

The MIA module performs the function of data bus transmission encoding/ decoding for communication between the SCU and the MDM controlling unit.

The SCU, which is the heart of the MDM, is in charge of all the MDM's internal operations and the controlling of data flow between the serial data bus and the IOM's.

The A/D converter module converts analog systems input information to digital information when it is returned to the MDM's controlling unit. The IOM's are the link between the SCU and the shuttle subsystems. Information transfers between the SCU and the IOM's are done on an internal data bus, while information transfers between the IOM's and the subsystems are done on individually dedicated lines.

Nine different types of slot interchangable IOM's can be used in any combination to establish a required MDM configuration. The nine types of orbiter MDM I/O modules are listed below:

IOM's	Range
1. Discrete input low (DIL)	0 and 5 V dc
2. Discrete input high (DIH)	0 and 28 V dc
3. Discrete output low (DOL)	0 and 5 V dc
4. Discrete output high (DOH)	0 and 28 V dc
5. Analog input differential (AID)	-5.12 to +5.11 V dc
6. Analog input single-ended (AIS)	-5.12 to +5.11 V dc
7. Analog output differential (AOD)	-5.12 to +5.11 V dc
8. Serial input/output (SIO)	-5 and +5 V dc
9. Tacan - a special purpose module for tacan and radar altimeter inputs and outputs (TAC)	0 and 5 V dc

The two power supply modules (frequently referred to as the core power supplies) provide redundant power to all modules in an MDM. They also supply power to each of the individual IOM power supplies.

One MIA module, one SCU module, and one A/D converter module make up the "front end" of an MDM. Frequently the term "MDM port" is used, and such reference may be to the entire "front end" of the MDM or to the MIA module only, which can sometimes lead to confusion. An orbiter MDM can be controlled through either of the two "front end" ports, but not simultaneously. If a hardware failure occurs in one port or on one data bus, the other port can be manually selected (via crew SPEC item entry) in an attempt to recover the data interface. The IOM's are common to both ports;

therefore, total redundancy of IOM's does not exist. However, there is a small portion of redundant electronics, the IOM select logic, that is port-dependent. If a failure were to occur in the redundant portion of the select logic, port moding may recover the data path. IOM circuits are designed so that, if a circuit failure were to occur, the failure would remain isolated to a bit, a card, or to an MDM. Information flow on the data bus is in the form of serial 28-bit words, transmitted at a one bit per microsecond rate with a 5 to 6 microsec gap between words. Each word can contain MDM command information, an analog signal, 16 discrete signals, or 16 bits of serial digital information. Note that the orbiter GPC's maintain a word gap of 5 microsec between command words/command data words. MDMs, however, maintain a word gap of 6 microsec between response data words.

Command word information is retained by the SCU in a register file to be used for mode control (what the MDM will do) and to initialize the MDM for processing of the command data word and data words that follow. A command data word contains the digital data that is serially gated to the IOM where it can be used by the subsystem.

The MDM has two modes of operation for both input and output transactions; these are referred to as direct and indirect modes. The mode to be used by the MDM is defined by a four-bit field in the command word. The direct output mode consists of a command word followed by from 1 to 32 command data words. The direct input mode consists of one command word initiating the transmission of from 1 to 32 non-built-in test equipment (BITE) response words, or from 1 to 64 BITE MDM response data words.

The indirect mode allows more than 32 words of information to be sent to, or received from, the shuttle subsystems with one command word. By using the 512-word PROM, up to 512 command data words can be output or up to 1024 BITE (512 non-BITE) response data words can be input in one transaction.

The MDM, when requested, will convert the subsystem data to serial digital form and send it back to the requester. The return information, formatted into a 28-bit response data word, can contain 1 analog signal, 16 discrete signals, or 16 serial digital bits of information.

7.1.2 System

There are seven basic types of orbiter MDM's on every flight (FF, FA, PF, LF, LA, OF, and OA). Figure 7-2 shows the data bus network interface and the 23 space shuttle MDM's. Nineteen of the MDM's are located onboard the orbiter. The other four MDM's are located in the SRB's.

The 19 orbiter MDM's are as follows:

A. FF 1, 2, 3, and 4 (flight-critical forward)

B. FA 1, 2, 3, and 4 (flight-critical aft)

C. PF 1 and 2 (payload forward)

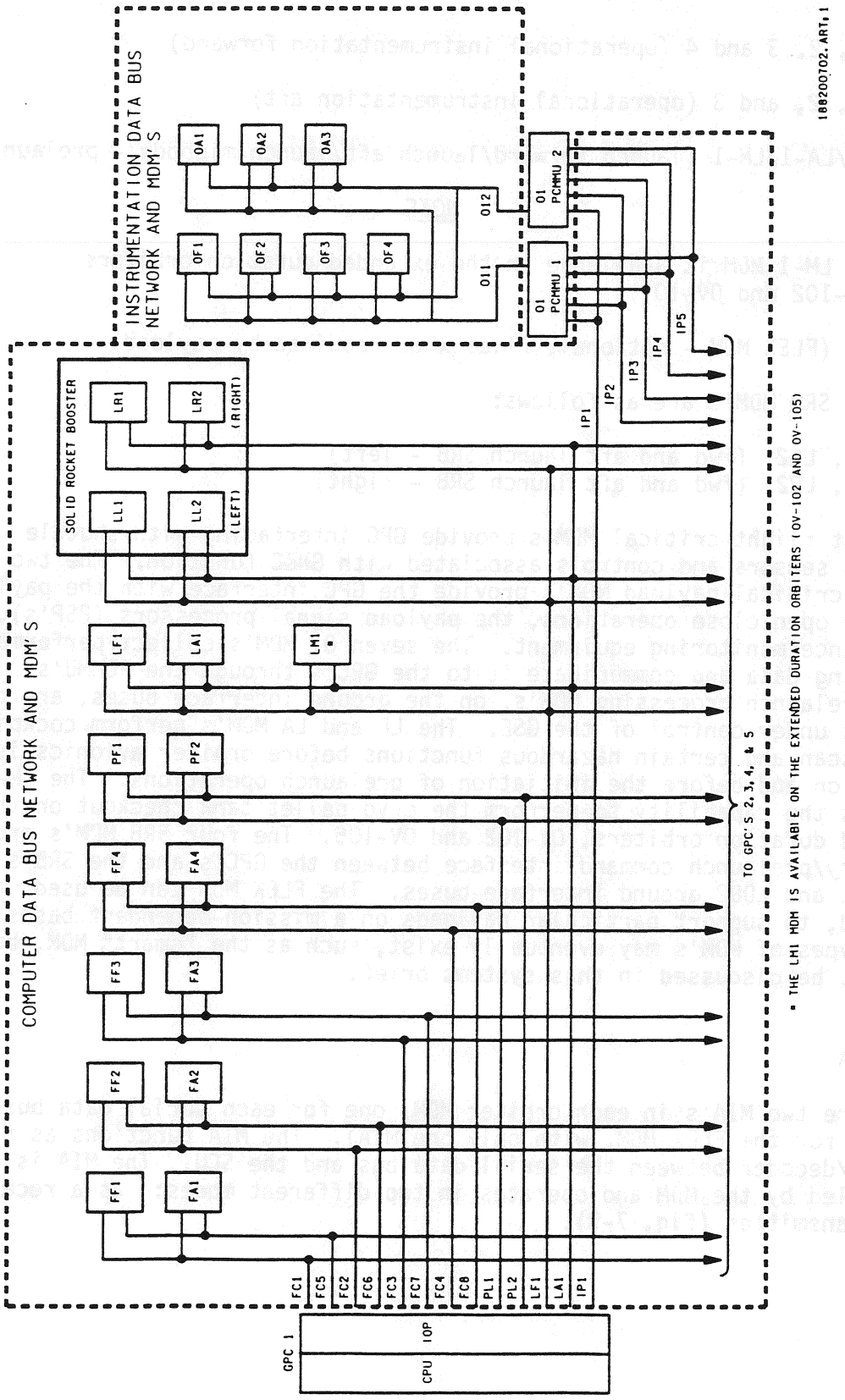


Figure 7-2.- Shuttle vehicle MDM's.

D. OF 1, 2, 3 and 4 (operational instrumentation forward)

E. OA 1, 2, and 3 (operational instrumentation aft)

F. LF-1/LA-1/LM-1 (launch forward/launch aft/launch midbody - prelaunch)

NOTE

The LM-1 MDM is available on the extended duration orbiters (OV-102 and OV-105).

G. FLEX (FLEX MDM - optional, flies when required by payload)

The four SRB MDM's are as follows:

LL1, LL2 (fwd and aft launch SRB - left)
LR1, LR2 (fwd and aft launch SRB - right)

The eight flight-critical MDM's provide GPC interfacing with shuttle avionics sensors and controls associated with GN&C function. The two mission-critical payload MDM's provide the GPC interface with the payload bay door open/close operations, the payload signal processors (PSP's), and performance monitoring equipment. The seven OI MDM's collect performance monitoring data and communicate it to the GPC's through the PCMMU's. The three prelaunch processing MDM's, on the ground interface buses, are command decoders under control of the GSE. The LF and LA MDM's perform cockpit switch scan and certain hazardous functions before orbiter avionics is powered on and before the initiation of prelaunch operations. The LM-1 MDM provides the capability to perform the cryo pallet tank checkout on the extended duration orbiters, OV-102 and OV-105. The four SRB MDM's provide a telemetry/prelaunch command interface between the GPC's and the SRB's on the and LDB1 and LDB2 ground interface buses. The FLEX MDM can be used, as required, to support particular payloads on a mission-dependent basis. Other types of MDM's may eventually exist, such as the "smart" MDM, but they will not be discussed in this systems brief.

7.2 MIA

There are two MIA's in each orbiter MDM, one for each serial data bus port (except for the FLEX MDM, with only one MIA). The MIA functions as the encoder/decoder between the serial data bus and the SCU. The MIA is controlled by the MDM and operates in two different modes: as a receiver or as a transmitter (fig. 7-3).

7.2.1 Receive Mode

The MIA receive mode is initiated when the SCU enables the decoder receiver. The receiver must be activated at least 1 microsecond before a valid sync can be received from the data bus to assure proper sync detection. The MIA will not attempt to decode any serial data bus information until a valid command or data word sync is detected. When a valid sync is detected, during the first 3 ms of a data word, the MIA will convert the next 25 bits of serial Manchester encoded information into serial non-return-to-zero (NRZ) code one bit at a time and make it available to the MDM. Each bit remains on the NRZ data bus for 1 ms. A clock pulse from the MIA will enable the SCU to sample each bit while it is on the NRZ data bus. After all 25 bits have been clocked through, the MIA generates a pulse marking the end of the word. At this time, the MDM will sample the three MIA error indicators: parity error, bit count, and nonvalid Manchester. A parity error would indicate that the odd parity of the received data word was not correct. The bit count error would indicate that less than 25 data bits were received. The nonvalid Manchester bit is set if any of the received data bits do not conform to proper Manchester code. Any one of the MIA-detected errors will cause the MDM to stop processing on that word and all the rest of the data words in the transaction will be discarded. The incoming data error bit in the BSR will also be set indicating that an error has occurred.

7.2.2 Transmit Mode

To enable the MIA transmit mode, the SCU enables the encoder transmitter. The transmitter must be enabled at least 1 microsecond before data transmittal to the MIA begins. The SCU transmit control logic will indicate to the MIA whether a command word sync or a data word sync is to be generated. The MIA's sync generator will generate the proper word sync, which is converted from NRZ to Manchester. The signal is then sent to the transmitter for shaping and amplification, and on to the data bus through a transformer. The first bit of data (NRZ encoded) transmitted by the MIA must be available to the MIA before the sync pulse is complete. The MIA samples the NRZ data and converts it to Manchester data, which is then transmitted to the data bus at a rate of one bit every microsecond. When the MIA is ready to accept the next bit for conversion, it sends a transmit shift clock signal to the SCU. After all 24 data bits have been transferred and converted, the MIA will generate an odd parity bit (based on the 24 data bits) and transmit it in Manchester form as the last bit in the command or data word. The transmission timing of individual data words is controlled by the MIA, but the timing of gaps between transmitted data words is controlled by the SCU, which provides the data and control signals in the proper timing relationships.

7.3 SCU

The SCU is the heart of the MDM, controlling all the internal MDM operations by executing microprograms contained in the control store memory. There are two SCU's in each orbiter MDM, one on each port (one SCU for SRB and FLEX MDM's). Only one SCU can be active at a time. The SCU controls the flow of data and provides timing to accomplish this task.

A SCU is made up of the following parts:

- Microprogram control
- MIA interface logic
- Internal bus data flow control
- SCU OUT bus/IN bus control number
- IOM interface
- Cyclic compare logic
- Power interrupt/monitor logic
- Systems clock

The current MDM only has one SCU OUT bus for all IOM's and an SCU IN bus for each IOM. The EMDM has an SCU OUT bus for each IOM which participates in both input and output data transfers.

7.3.1 Microprogram Control

All operations in the SCU are directed by the microprogram control. The microprograms are stored in the control store memory (512 words, 40 bits per word) PROM. Microinstructions are executed at a rate of one per microsecond. The 40-bit microinstruction words are divided into fields that control various SCU operations. The next microinstruction will be determined by the instruction decoder and memory address select and its inputs; the literal field of the current microinstruction, forced program entry discretes, the mode/IOM type branch field from the register file, and conditional instructions.

The conditional instructions can be either sensed or set and are contained in the condition field of the control store memory word. The 14 sensed conditions do checks on particular events at specific times and can modify the microinstructions based on these conditions. The 28 set conditions may be set one at a time by the microinstruction program for controlling internal operations of the SCU and also for recording events that occur while the microinstructions are being executed.

7.3.2 MIA Interface Logic

The MIA interface receiver sync logic receives serial command word and serial command data word information from the MIA and synchronizes it with SCU timing. Verification tests are performed on the incoming words for the following items:

- A proper sync code on the incoming serial word: command sync on the first word in a transaction, data sync on all the subsequent words
- An MDM address that compares with the hard-wired MDM address
- A check of the MIA indications: proper Manchester code, bit count, and odd parity
- A proper gap time between the preceding word and the incoming data word
- A proper SEV (101) check bit pattern on incoming command data words

If the incoming word passes these tests, an additional check is performed to make sure the other SCU in the MDM is not already busy. If the other SCU is not busy, power is then applied to the memory control logic and processing of the command or data word can begin.

Table 7-I indicates the bit fields that form the command words for the direct and indirect modes of data transfer.

TABLE 7-I.- COMMAND WORD

Direct mode		Indirect mode	
Bits		Bits	
1-3	Sync (removed at MIA)	1-3	Sync (removed at MIA)
4-8	MDM address	4-8	MDM address
9	Spare	9	Spare
10-13	Mode control field	10-13	Mode control field
14-17	Module address	14-22	PROM starting address
18-22	Channel address	23-27	Number of instructions
23-27	Number of words (channels)		

When the MIA receives a valid sync pattern (first 3 microseconds of a word), the other bits are serially shifted to the receiver sync logic circuitry five-bit shift register. This register takes the incoming serial information arriving at a 1 MHz rate, and transfers it to the five-bit receiver bus. When bit 8 has been serially shifted into the register, the incoming MDM address is shifted in parallel to the MDM address field decoder/compare logic where it is compared with the hard-wired address. If the address com-

pares and the MIA detected a command sync, the SCU control store memory is powered up and microinstruction execution will begin. If the address does not match, no further processing of the word will be done.

When bit 13 is clocked into the register, the mode control field (bits 10 to 13) is transferred to the parallel bus and then to the register file where it will be available for later use. The mode control field specifies what the MDM will be doing with the transaction and where the remaining fields of information are to be stored.

The response sync MIA interface logic synchronizes the 24 bits of response data word information to the MIA timing and enables the MIA transmitter.

7.3.3 Internal Bus Data Flow Control

This portion of the SCU contains the register file, counters, sequence read-only memory (ROM), BITE status register (BSR), the MDM wired address, and the SCU internal bus.

The register file consists of eight five-bit registers. For direct mode operations, one command word followed by one command data word, the file register holds the mode control, the IOM address, the channel address, and the number of words to be transferred fields. For the indirect mode operations, when the sequence memory ROM is used, the register file holds the sequence memory address and the number of sequence memory instructions to be executed (maximum of 16). One of the file registers contains bits 12 to 16 of the BSR test bits.

The five-bit upper and four-bit lower up/down counters are controlled by SCU microinstructions. They are used in the following ways.

- Counting down the number of words to be transferred
- Incrementing the channel address
- Holding and incrementing the nine-bit address for sequence memory (PROM) execution
- A conditional branch register (counter underflow)

The sequence ROM is a 512-word, 16-bit memory. The first 16 words are allocated to the IOM module configuration and are thus specified by one of the 7 basic orbiter types; FF, FA, PF, OF, OA, launch, and SRB. Each of the remaining 496 memory words resembles a command word. The contents of the memory are alterable only by wiring changes. Once a memory is mounted in the SCU, its contents cannot be altered.

The memory contains a number of fixed programs. Each program sequence is made up of a number of command words. The maximum program length is 32 command words. To start a sequential program, the first command word sent to the MDM by the controlling unit addresses the ROM. It specifies the in-

direct operation mode and the starting ROM address. After the first command word, the ROM command words that make up the program are controlling the MDM's information transfer.

The BSR file is a 16-bit register that holds error conditions detected by the real-time BITE. Testing is performed continuously during both input and output transactions. Each SCU has its own BSR. When an error condition is detected, it is set in the BSR of the active SCU. There is one exception. During the power-up and power-down sequences, the BSR power interrupt bit in both SCU's will be set.

The BSR is reset every time it is read. It is also reset when a command word is received by the MDM specifying a master reset. A master reset also resets all the outputs of the IOM's.

The MDM's wired address is used by the address check logic to do an address compare on all serial words on the data bus. If a word on the bus has a valid sync code and an address that compares with an MDM's wired address, that MDM will continue processing that word. If the addressing does not compare, the MDM will not continue processing of that word. The wired address is also used when a response data word is being built for return to the MDM's controlling unit. (For information on the wired address plug, see section 7.9.1.)

The SCU internal bus is a five-bit parallel bus that supports the transfer of data between the different devices in a SCU. The various bus interfaces are enabled whenever data are put onto and taken off of the bus and are controlled by the microprogram instructions.

7.3.4 Out Bus/In Bus Control

The out bus/in bus control is the pathway data takes going from the SCU to the output IOM's and from the input IOM's to SCU.

The current MDM has an SCU serial out bus multiplexer. This multiplexer is selected by microprogram control when outputting data from the SCU internal bus. It also has logical one and zero output sources that are used when generating the task bit and check bit codes for building response data words.

On the other hand, the EMDM has an SCU serial out bus decoder which provides the same function as the current MDM's serial out bus multiplexer. However, with each IOM having its own serial out bus it minimizes the chances that erroneous data is sent to the wrong IOM. See the EMDM Enhancement section of this systems brief.

The SCU in bus control formats the response data words when response data come to it from the IOM's. Each IOM in the current MDM has an SCU in bus. The EMDM uses its SCU out buses to transfer the data from the IOM's to the SCU. A logical one and zero are used when building the SEV (101) pattern.

The SEV information is checked by the SCU in bus control for each response data word built by the MDM.

7.3.5 IOM Interface

The IOM interface selects which IOM card or analog channel will be interfaced. It checks that the IOM has been selected with a valid channel address and task code, that the check bits are correct, and that the requested operation has been performed. The digital data are returned to the SCU through the IOM interface and analog data are returned to the SCU through the A/D converter (current MDM). The EMDM has an upgraded version of the IOM interface that performs the same functions as the current MDM interface. However, the upgraded version reduces the possibility of selecting the wrong IOM. See IOM Mapping under the EMDM Enhancement section of this systems brief.

7.3.6 Cyclic Compare Logic

The cyclic compare logic allows for the real-time monitoring and test of the execution of all SCU microinstructions. During the execution of each microinstruction, a total of 46 control points (defining the present SCU status) is reduced by the cyclic matrix to an 8-bit number. A running sum of these numbers (one per microinstruction) is maintained, and approximately every tenth microinstruction is compared against a sum-check constant. The sum-check constant is contained within the 40-bit microinstruction field of the microinstruction presently being executed. A comparison indicates the validity of all SCU operations since the previous sum check. A noncomparison results in an immediate reset of both the MIA and SCU, truncating any operation in progress. The reset condition can only be removed by removal and reapplication of power to the MDM.

Cyclic sum checks are performed a minimum of three times for each sequence of microinstructions associated with the processing of each command, command data, or response data word. Every microinstruction executed by the SCU forms a part of some sum check. A failure of one SCU by the cyclic comparison tests leaves the operational capabilities of the redundant SCU unaffected.

7.3.7 Power Interrupt/Monitor Logic

The power interrupt/monitor logic will issue a dead stop clear (DSC) causing an immediate reset of the MIA and SCU for a low SCU logic voltage level or when a cyclic error is detected by the cyclic compare logic.

The control store memory power enable logic is reset and the S-bit (power transient) is set by the power interrupt/monitor logic circuitry when a low bus voltage is detected by the energy level valid monitor in the power supply. This logic is also used to control the power supply BITE tests.

7.3.8 Systems Clock

Each SCU has a 16-MHz crystal oscillator that provides all the timing signals for the MDM. All the operations in the MDM are controlled by the system clock of the active SCU.

The following timing signals are for the current MDM:

A. 16-MHz - MIA

B. 8-MHz - Serial IOM's

C. 1-MHz - Other IOM's and SCU operations

The following timing signals are for the EMDM:

A. 4-MHz and 2 MHz - SCU operations (includes portions of the PROM and MIA interface logic)

B. 1-MHz - All IOM's and the A/D module

7.4 A/D CONVERTER

The A/D converter module converts analog inputs from shuttle subsystems into digital information for transmission to the MDM's host unit. All analog input modules are connected to the A/D converter by their own separate analog bus. This module converts analog inputs, ranging from +5.11 to -5.12 V dc, into 10 digital binary bits in two's complement form (least significant bit = 10 mV).

7.5 INPUT/OUTPUT MODULES

Each orbiter MDM has 16 IOM's of 6 different basic types that interface the MDM with the shuttle subsystems. Each flexible MDM and SRB MDM has eight IOM's.

7.5.1 Discrete Input Module

The discrete input module converts up to 48 parallel inputs into 3 serial, 16-bit data words. Logical zero = 0 V dc. A discrete input low (DIL) logical one = +5 V dc. A discrete input high (DIH) logical one = +28 V dc.

7.5.2 Discrete Output Module

The discrete output module converts three 16-bit serial digital command words into 48 parallel discrete signals for use by orbiter systems hardware. A discrete output high (DOH) logical one = +28 V dc. A discrete output low (DOL) logical one = +5 V dc, logical zero = 0 V dc.

7.5.3 Analog Input Modules

Analog input modules condition and multiplex analog inputs ranging from +5.11 V dc to -5.12 V dc, from orbiter systems hardware for A/D conversion. The resulting digital words (one word per channel) can be transmitted to the MDM's controlling unit when requested. Multiplexing at the A/D converter prevents a failure of one analog input module from affecting other analog input modules.

The analog input differential module (AID) receives 16 pairs of signal inputs (16 channels). The maximum voltage differential of a signal pair is 5.12 V dc.

The analog input single-ended module (AIS) receives up to 32 single ended dc inputs (32 channels). The minus (-) input signal leads are internally grounded.

7.5.4 Analog Output Differential (AOD) Module

The AOD converts 10-bit digital commands (least significant bit = 10 mV) into analog outputs (+5.11 V dc to -5.12 V dc range) for use by the orbiter system hardware. Up to 16 pairs of analog output signals can be provided.

7.5.5 Serial Input/Output (SIO) Module

The SIO provides four separate bidirectional, half duplex, SIO channels for interface to the Space shuttle subsystems. One channel can be active at a time. Information transfer is in 20-bit words, 1 ms per bit, Manchester II format (logical one = 10, logical zero = 01). The 20-bit word format is: 3 sync bits followed by 16 data bits and 1 parity bit.

The SIO words are synchronized with and originate from command data words to the MDM. SIO words from the IOM are formatted and transmitted as response data words. One 20-bit serial I/O word's information is equivalent to a 28-bit command data word or response data word.

7.5.6 Tacan/Radar Altimeter (TAC) Module

The TAC interfaces with the tacan control panel, the tacan unit, and the radar altimeter.

- Tacan control panel - Fifteen low level discrete signals (0 or 5 V dc) from the control panel to the IOM indicating switch closures. The 5 V dc is provided by the IOM.
- Tacan unit - This interface consists of four (0 or 5 V dc) discrete signals, two inputs and two outputs; and two simplex serial data channels, one for data flow from the MDM to the tacan unit and one from the tacan unit to the MDM.

- Radar altimeter unit - This interface consists of one (0 or 5 V dc) discrete signal, three differential signals, and one simplex serial data channel.

7.5.7 Input/Output Module Configuration

The input/output module configuration for the MDM's are listed in table 7-II.

7.6 POWER SUPPLY

Figure 7-4 shows how the orbiter and SRB MDM's receive power. Each MDM is redundantly powered by two of the three main buses. Control bus power activates a corresponding RPC. One of the two control buses on each FF, FA, and PL MDM is sent to the ground via telemetry. This enables the ground to monitor the MDM power switch status.

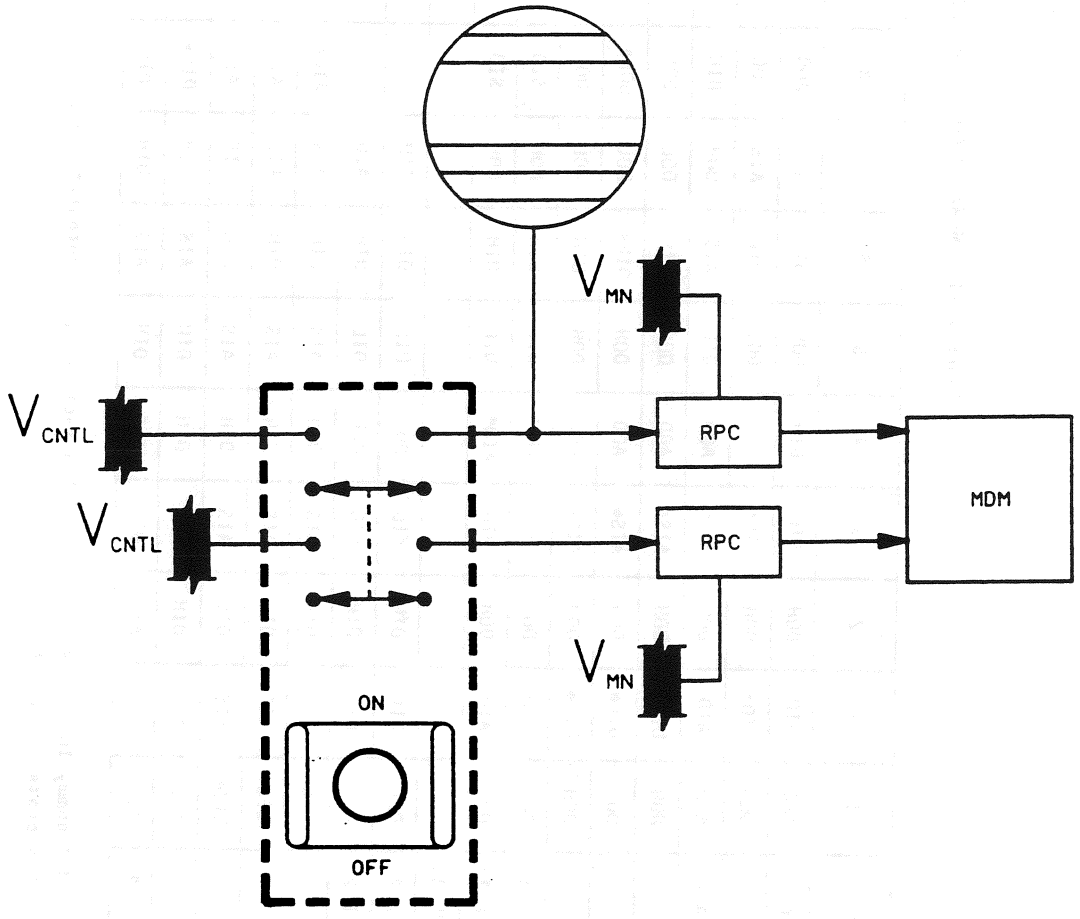


Figure 7-4

TABLE 7-II.- MDM/IOM CONFIGURATION

MDM	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
FF 1-FF 2	TAC	AID	DOH	SIO	DIH	DOL	DIL	AIS	AOD	DIH	DOH	SIO	DIH	DOL	AID	DIL
FF 4	TAC●	AID*	DOH	SIO●	DIH	DOL	DIL	AIS	AOD	DIH	DOH	SIO	DIH	DOL	AID	DIL
FA 1-FA 4	AOD	AID	DOL	DIH	AOD	DIL	AIS	DOH	DIH	AID	DOL	DIH	DOH	DIL●	AIS	DOH
LF 1	DOH	DIL●	DOH	AIS●	AOD	DOH	DIH	DOL	DOL	DOH	DIH	DOH	AID●	DIH●	DOH	DIL●
LA 1	DOH	DIL●	DOH	AIS●	AOD	DOH	DIH●	DOL	DOL●	DOH	DIH●	DOH	AID●	DIH●	DOH	DIL●
LM 1	DOH	DIL●	DOH	AIS●	AOD	DOH	DIH	DOL	DOL	DOH	DIH	DOH	AID●	DIH●	DOH	DIL●
PF 1(PL 1)	DOL	AID	DOH	DIH	AID*	DIL	DIH	DOH	SIO	DIH	DOL	AID	AOD●	DIL	DOH	SIO●
PF 2(PL 2)	DOL	AID	DOH	DIH	AID*	DIL	DIH	DOH	SIO	DIH	DOL	AID	AOD	DIL	DOH	SIO●
PM 1(PL 3)																
LL 1,LL 2	DOL	DIL	DIH	AID	DOL	DIL	DIH	AID								
LR 1,LR 2	DOL	DIL	DIH	AID	DOL	DIL	DIH	AID								
OF 1	SIO	AIS	DIL	AIS	DIH	AIS	DIH	AIS	AID●	AIS	DIL	AIS	DIH	AIS	DIH	AIS
OF 2	SIO	AIS	DIL	AIS	DIH	AIS	DIH	AIS	AID	AIS	DIL	AIS	DIH	AIS	DIH	AIS
OF 3	SIO●	AIS	DIL	AIS	DIH	AIS	DIH	AIS	AID	AIS	DIL	AIS	DIH	AIS	DIH	AIS
OF 4	DIL	AIS	DIH	DIL	DIH	DIH	AIS	DIH	DIL●	AIS	DIH	AIS	DIH	DIH	AIS	DIH
OA 1-OA 3	AIS	DIH	AIS	DIL	AIS	DIH	AIS	DIH	DIL	AIS	DIH	AIS	DIH	AIS	DIL	AIS

+ and * dummy IOM's used in GPC I/D transactions (+ = output/* = input).

● IOM's present but not used.

7.6.1 Core Power Supply

Each MDM has two redundant core power supplies. Each core power supply provides power to one port of the MDM, as well as to each of the individual IOM power supplies in the MDM. A port consists of one MIA, SCU, and A/D converter. Figure 7-5 illustrates the power distribution in the MDM. The SRB MDM core power supply is unique in that it does not utilize the 10-volt regulator.

The figure also illustrates the cross-strapping that occurs between the second-stage regulators of the core power supplies. This cross-strapping allows a second-stage regulator to draw power from the first-stage regulator of either core power supply. For example: port 1, powered through power supply 1, can also draw power from power supply 2. Port 2 can also draw power from either core power supply. Therefore, if a main bus is lost, either port can draw power from the remaining main bus. Power supplied to the IOM's is also cross-strapped, as discussed in the next section.

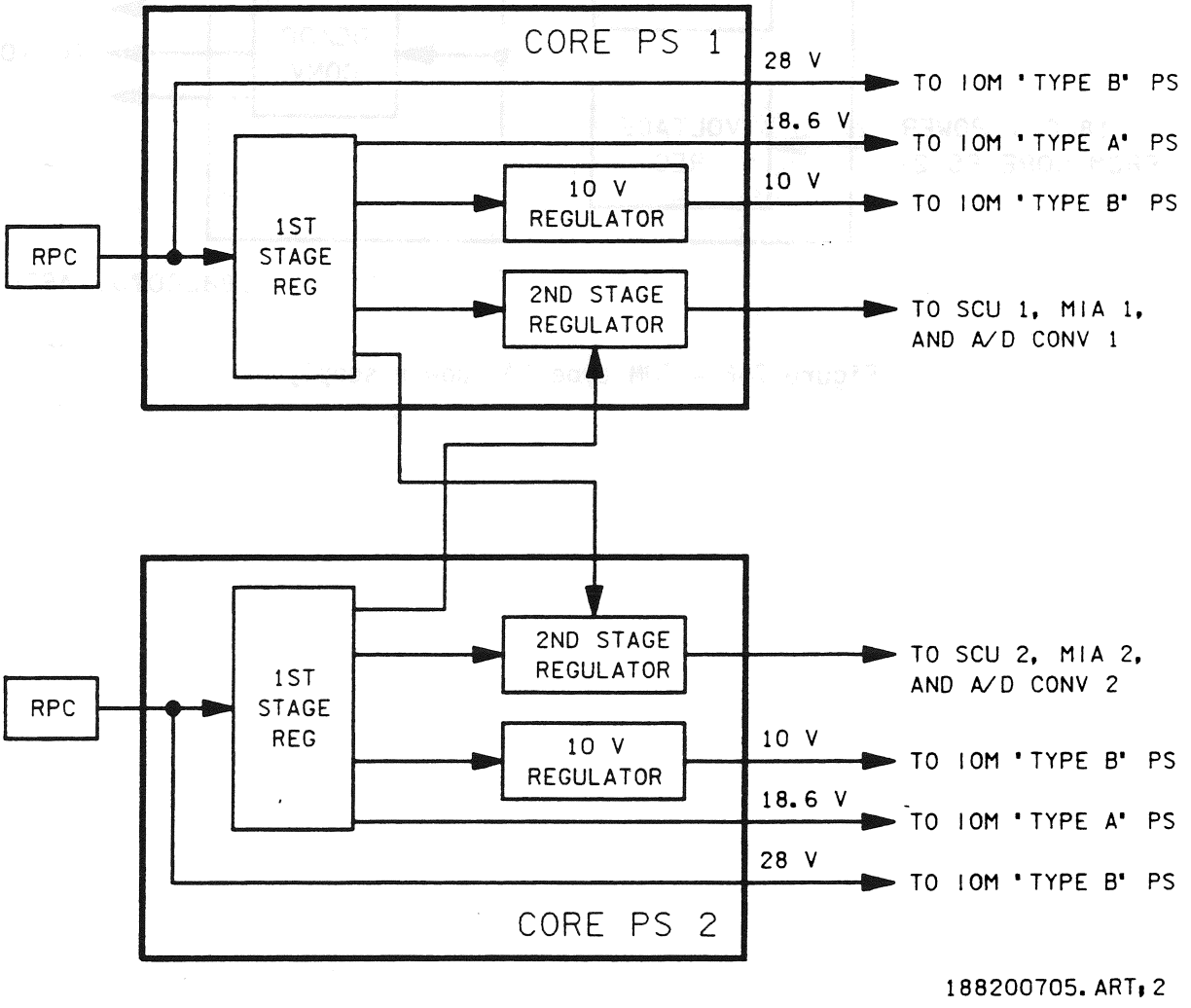


Figure 7-5.- Power distribution in MDM.

7.6.2 IOM Power Supply

Each IOM in the MDM has its own power supply. There are two basic types of IOM power supplies: type A and type B. The SRB IOM power supply is slightly different type, but is is functionally the same as type A.

The IOM type A power supply is illustrated in figure 7-6. Power is input to it from the first-stage regulator of each of the core power supplies. These redundant power buses are cross-strappped, allowing the IOM to draw power from either main bus via the corresponding core power supply. The IOM type A power supply is used by the following types of IOM's: AIS, AID, AOD, DOL, SIO, and TAC.

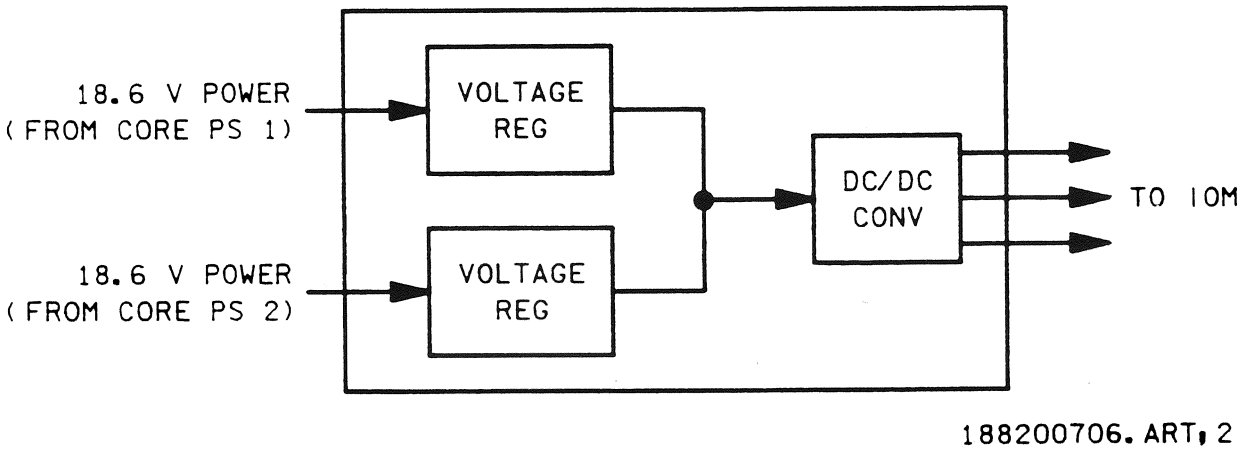


Figure 7-6.- IOM type "A" power supply.

The IOM type B power supply is illustrated in figure 7-7. This power supply receives 10-volt and 28-volt power from each of the core power supplies. Each pair of power buses (the 10-volt pair and the 28-volt pair) are cross-strapped, allowing the IOM to draw power from either main bus via the corresponding core power supply. The IOM type B power supply is used by the following types of IOM's: DIL, DIH, and DOH.

In the current orbiter MDM, the 16 IOM power supplies are located on two cards, with eight power supplies per card. In an orbiter EMDM, the IOM power supplies are located on their corresponding IOM card. In an SRB MDM, the eight IOM power supples are also located on two cards, with four power supplies per card. The flexible MDM's have the IOM power supplies located on the corresponding IOM card. Although IOM power supplies may be physically located on the same card, they are functionally independent.

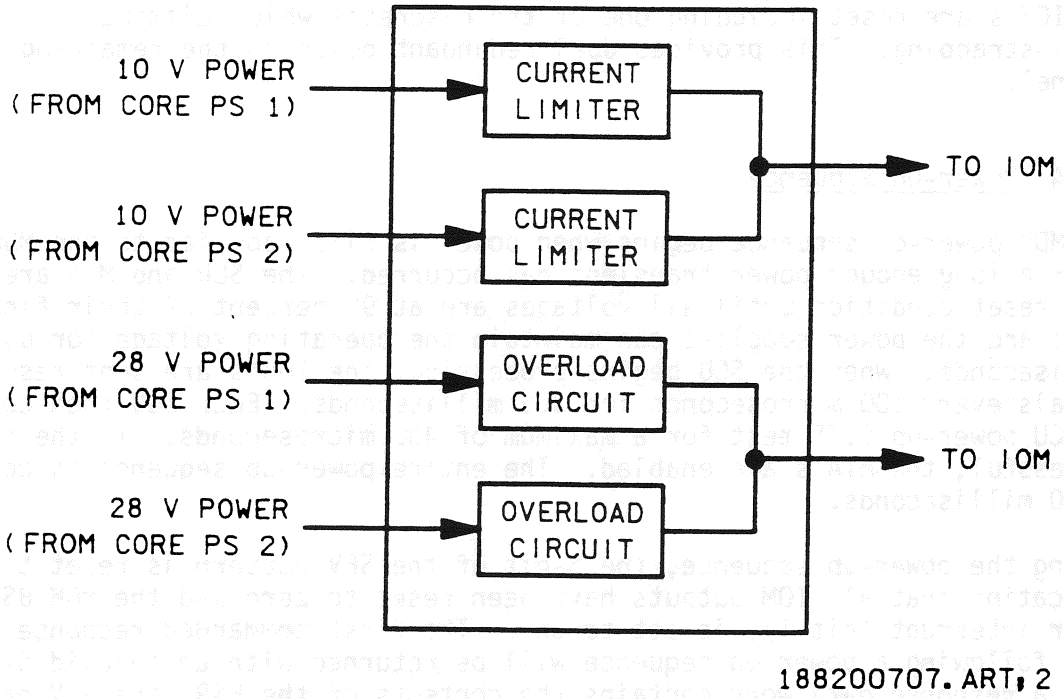


Figure 7-7.- IOM type "B" power supply.

7.6.3 SRB MDM Power Separation

The design of the power distribution in the SRB MDM's has a unique feature that is not present in the orbiter or flexible MDM's. This feature is known as power separation and effectively disables the cross-strapping of core power supplies and IOM power supplies.

The power separation was designed into the SRB MDM's due to a single-point failure discovered in the SRB MDM. A dead short in one of the IOM power supplies would result in a momentary voltage drop to all other power supplies in that MDM because of cross-strapping. This voltage drop would cause all output cards to reset their outputs, just as a power cycle would. This is not a problem for the orbiter MDM's because the GPC will immediately refresh the state of the outputs. However, the SRB MDM outputs are not refreshed by the GPC's. On aft SRB MDM's gimballing would be lost and eventually the vehicle would be lost.

The redesigned SRB MDM uses an output discrete from two of the IOM's to eliminate the cross-strapping power supplies in that MDM. These discretes are set during prelaunch by ground commanding. In this case, if a dead short occurs in one of the IOM power supplies, only one port and half of the IOM's would see the drop in voltage. Therefore, only one channel in the MDM would be lost. Also, when the voltage drop occurred, the ouputs on half of the IOM's are reset including one of the discretes which eliminates cross-strapping. This provides dual redundant power to the remaining channel.

7.6.4 Power-On Sequence

The MDM power-on sequence begins when power is first applied to the MDM or after a long enough power transient has occurred. The SCU and MIA are held in a reset condition until all voltages are at 90 percent of their final value and the power supplies can maintain the operating voltage for up to 2 milliseconds. When the SCU begins processing, the IOM's are sent reset signals every 100 microseconds for 3.5 milliseconds. Each SCU then conducts an SCU power-up BITE test for a maximum of 400 microseconds. If the test is successful, the MIA's are enabled. The entire power-up sequence is complete in 10 milliseconds.

During the power-up sequence, the S-bit of the SEV pattern is reset to zero, indicating that all IOM outputs have been reset to zero and the MDM BSR power interrupt (bit 1), is set to one. The first commanded response data word following a power-up sequence will be returned with an invalid S-bit. When a response data word contains the contents of the BSR, the SEV pattern is forced to valid "101" pattern ensuring the receipt of the BSR information by the controlling unit.

7.7 BUILT-IN TEST EQUIPMENT (BITE)

7.7.1 BITE Overview

The MDM SCU has BITE that performs error detection functions. BITE can test the SCU itself, the core power supplies, A/D converters, IOM's, SCU PROM, and the BSR. Detectable errors encountered during BITE testing and the status of the BITE test being processed are stored in the 16 bits of the BSR file.

The majority of the BITE testing capability in the MDM requires specific commands to start the testing. Presently, the only software that contains BITE test commands is available in OPS 9 and is normally used during the preflight and postflight timeframes. Some real-time BITE testing is performed during input and output transactions. Errors detected during real-time BITE testing will set appropriate bits (1 through 10) in the SCU's BSR.

A detected problem on an output transaction will remain transparent to the MDM's controlling unit until a command word sent by the controlling unit to the MDM requests the return of the BSR contents. Any bits set in the BSR will inform the controlling unit of errors that have been detected since the last time the BSR was reset.

A problem detected during an input transaction will be indicated to the controlling unit immediately. This is done by an invalid SEV pattern in the response data word being processed when the error is detected. The appropriate bit or bits will be set in the BSR also. The SEV pattern returns to the normal 101 pattern on subsequent data words if no more errors are detected.

7.7.2 BSR File

The 16 BSR bits are described below.

Bit 1 Power Interrupt. This bit is set in only two instances: during the power-up sequence and 100 μs prior to MDM shutdown. The bit indicates that all analog and discrete outputs have been, or are about to be, reset to 0 V dc.

Bit 2 Incoming Data Error. This bit indicates that the incoming command word or command data word was detected to have an error.

For command words (with valid MDM address) the error is either nonvalid Manchester code, too few bits, or bad parity. If the command word is part of a command data word message, subsequently appearing command data words are discarded and cause no additional bits to be set in the BSR.

For command data words, this bit is set for incorrect MDM address, nonvalid Manchester code, too few bits, improper SEV code, or parity error. The word is not processed. Subsequent command data words of the same message are also not processed.

Bit 3 Operation Requested on a Nonexistent Channel. This bit can be set during command data and response data messages. The occurrence of the error does not halt the command data or response data word message sequence.

The command word channel address field allows 32 channels to be addressed; however, not all IOM types have 32 channels. The above bit is set if the channel address field specifies an address beyond the range of the selected IOM. If this error occurs, BSR bit 4 is also set (unable to transfer data to/from selected IOM).

For command data word messages, subsequent command data words are accepted if the channels are subsequently correctly addressed. During indirect mode processing (use of SCU PROM), this implementation will cause outputs to be accepted and issued by the remaining IOM's of the PROM sequence.

For response data word messages, an operation requested on a nonexistent channel does not prevent a serial data bus word from being issued to IOP, PCM, or GSE units. The data contained within the response data word are invalid, however, and the V-bit is cleared to a logical 0 to indicate this condition. Processing continues. V-bit decisions are made on a response data word basis thereafter.

BSR bit 3 is set during command data and response data word times. If the IOM is operating properly, the actual error occurred in the preceding command word. This command word had an improper value in one or more of the following command word fields: module address, starting channel address, or number of words to be transmitted received.

Bit 4 Unable to Transfer Data to/From IOM. This bit can be set during command data and response data messages. Processing can continue after the error condition. The bit is set if the IOM fails to reply to the SCU with a data transferred (operation completed) signal. For response data words, the V-bit is cleared for the word that encountered the error. Thereafter, V-bit decisions are made word by word.

Bit 5 Too Many Words in Last Message. This bit can only be set during command data word messages. The first extra command data word of the command data word message causes the bit to be set. The extra command data word must fall within the allowable gap time limits (4 to 11 μs); otherwise BSR bit 8 is set (illegal mode), indicating receipt of a command data word with no previous command word.

Additional command data words of the message are discarded and cause no additional bits to be set in the BSR.

Bit 6 Last Command Not Completed. This bit is set if a new command word is received on the same bus prior to the completion of processing for the previous command word on that bus. The new command will be accepted.

Bit 7 Simultaneous Execution on Primary and Backup Data Bus. This bit is set if a command word is received by one SCU while the other SCU is busy. The condition will result in an immediate halt of processing

for both SCU's. The interrupted SCU does not wait to complete processing for the command data or response data word in progress at the time of the interrupt. Either SCU will continue processing when it receives a new command word.

Bit 8 Illegal Mode Commanded. This bit can be set by either a command word or a command data word.

For command words, the bit is set for an incompatibility between the mode control and module address fields; for example, an output operation is specified for an input module. The bit is also set if an SCU PROM program attempts to execute both input and output operations in the same program. The bit is set if one of the four spare modes is commanded. The bit is set if one of the first 16 PROM words classifies an IOM as spare and the SCU attempts to address this IOM.

For command data words, the error condition is detected if the MDM receives a command data word without a prior command word.

Bit 9 Internal Error Detected. When this bit is set, the message (command data or response data) in progress is terminated upon occurrence of the error. However, a response data word in transit at the time of the error is completed with the V-bit cleared to a logical 0. BSR bit 9 is set in four instances: parity error for any of the 512 words of the sequence memory PROM, the IOM reply line is a logical one prior to the channel present time slot, the A/D reply line is a logical one prior to the data transferred (operation completed) time slot, or the SCU IN bus check bits contain an invalid code.

Bit 10 Gap Time Error. The SCU monitors the gap time between command data words and between the command word and the first command data word. If an improper gap time is detected, the gap time error bit is set. All subsequent command data words are ignored until a new command word is received. The nominal proper gap between all words is 5.5 $\pm$ 0.5 μs. A gap time error is indicated when the gap is less than 4 μs or more than 11 μs.

Bit 11 Successful BITE Completion. This bit indicates that a successful self-test was performed.

Bits 12 through 16 BITE Test. This field indicates progression through the BITE tests. The field is only applicable for SCU, power supply and A/D BITE tests at all other times these bits stay reset to zero. They exist in one of the registers in the register file (section 7.3.3).

7.7.3 SCU BITE

During the power-up sequence, the SCU BITE test is automatically performed in each SCU. If the BITE test is successful, the SCU is released for normal MDM operations. If the SCU BITE test is not successful, the SCU will not

respond to any command directed to it on the serial data bus. The success or failure of the SCU BITE test on either SCU will not affect operation of the other SCU or the associated serial data bus.

During the power-up sequence, the following SCU BITE tests are performed:

- The MDM receiver sync addressing logic is checked to verify response to only one of the possible 32 addresses.
- The microprogram control store cyclic compare logic operation is verified by executing a microsequence with an incorrect cyclic check.
- The SCU counters are tested.

Bit 1 (power interrupt) is the only bit in the BSR that will be set by SCU BITE test during the power-up BITE sequence.

After the SCU is operational, the SCU BITE test can be commanded by the MDM controlling unit. Only the SCU connected to the serial data bus over which the command is issued will perform the self-test. The following five SCU tests will be performed along with the previously described three tests in the current MDM. Two extra SCU tests are performed in the EMDM which is also indicated below.

- The command data word SEV pattern detection capability is tested by generating invalid SEV patterns and verifying that the errors can be detected.
- The SCU PROM odd parity detection capability is tested by forcing bad parity.
- The SCU register file is checked for read, write, and addressing errors.
- The BSR bits 2, 3, 4, 5, 6, 8, 9, and 10 are set with firmware. If the SCU BITE is successful, bits 11, 12, 15, and 16 will also be set.
- The IOM select and SCU data out decoders within the two core gate arrays are each checked to ensure that only one of the 16 signals is active at a time (EMDM only).
- Sequence memory test table is checked for proper pattern (EMDM only).
- The SCU out bus, SCU in bus, and IOM reply lines are verified by the SCU communicating with each IOM and monitoring the response on the IOM reply lines.

After the test is complete, the MDM's controlling unit must request the contents of the BSR to determine the results of the test.

The power supply BITE test of the redundant power supply network can be initiated on either MDM port. The test takes 42.1 μs to complete. After the test is completed, a BSR read is required to get the test results.

7.7.4 Core Power Supply BITE

The core power supply BITE test is conducted in two parts. The SCU that received the power supply BITE test command word considers itself the primary SCU for the test. It also considers one-half of the power supply network as its primary power source. The first half of the test will test the backup power supply. The primary power source will be tested during the second half of the test. The test consists of perturbing the reference voltage on one power supply, forcing the other power supply to supply power to the entire MDM while the load currents are being monitored. A successful test requires each half of the power supply network to yield and assume the full power load. A failure of either side to accept full load will be indicated in the BSR. The EMDM has an added capability such that an IOM power supply failure can be isolated to a particular IOM which is indicated in the last four bits of the BSR.

7.7.5 A/D Converter BITE

The A/D converter BITE is initiated by a command word over the serial data bus (direct mode). Five tests are performed resulting in five response data words being returned to the requesting controlling unit. The five response words are for a full-scale negative value, a midscale reference negative value, a null value, a midscale reference positive value, and a full-scale positive value. The performance of the A/D BITE is verified by the MDM's controlling unit comparing the returned values with what they should be. The BSR contents are used to indicate validity of the A/D test so it also is returned to the controlling unit.

7.7.6 IOM BITE

A. Input

The performance of input IOM's is determined by the MDM controlling unit comparing the test response data words with constants. The constants are values that are predetermined for each channel of each IOM type.

B. Output

The MDM output test, referred to by KSC personnel as a "BITE test 4," will cause the GPC to issue a BITE test 4 command word (mode control field of 0111). It will return the outputs currently on the first AOD, DOH, or DOL card on the MDM addressed. The test can be continued for the rest of the output cards in that MDM.

7.7.7 SCU PROM BITE

For the SCU PROM BITE test, the GPC will issue a command word requesting the return of the first 16 words of the PROM in the MDM. If the return words compare with what the GPC expects, the test will continue with verification of the rest of the PROM. The other 496 PROM words are returned to the GPC and their contents summed 32 words at a time.

7.7.8 BSR BITE

For the BSR BITE test, the GPC issues a command word with a mode control field of 1110 followed by a command data word with all ones in bits 9 through 24. The GPC will then request a BSR read (mode control field of 1010) and check the return word to see if the BSR bits are all ones. The same steps are repeated with all zeros in the command data word before the test is complete.

7.8 SEV BITS

Three SEV bits, normally set to 101, are part of every response data word from an MDM. They are indicators of BITE and hardware detected errors that cause the data in the response data word to be suspect. The S, E, and V bits are described below.

S-bit - When reset to zero, a power transient has been detected by the MDM and all the IOM outputs have been reset.

E-bit - When set to one, a serial input/output card has detected a data problem (no data, improper sync, or invalid Manchester II data bits) or the PCMMU has detected a data problem (no OI MDM data).

V-bit - When reset to zero, the MDM has detected an internal problem causing the data to be suspect/invalid.

7.9 EMDM ENHANCEMENTS

7.9.1 Gate Array Logic

Through its enhancements, the EMDM provides greater reliability than its predecessor. The foremost enhancement was the use of gate array technology. Previously, the current MDM's used an abundance of hybrids which were a combination of digital and/or analog circuits that provided certain tasks. However, in the EMDM many of these digital hybrids were redesigned into gate arrays in order to "enhance long term producibility and reduce the parts count." The gate arrays chosen, the LL9000 series, can be easily upgraded to the LRH9000 series without hardware redesigns, thus ensuring that the EMDM's components will not become obsolete in the very near future.

7.9.2 IOM Power Supplies

Another enhancement the EMDM offers is individual power supplies mounted on each IOM. Figure 7-8 shows the difference between the power supply distribution between the current MDM and the EMDM. This feature eliminates two modules that exist in the current MDM, thus allotting more room in the EMDM chassis. In addition, this enhancement allows the BITE to determine specifically which IOM has failed. These IOM failures can be identified by the last five bits of the BSR. Therefore, faster troubleshooting can be done to the boxes, and this quickens the turnaround time of the orbiter for its next flight.

7.9.3 Additional Circuit Designs

The last EMDM upgrade is the additional circuit designs and microde changes to minimize erroneous data outputs that could have a sever impact on the orbiter avionic system. These enhancements include the SCU word wrap, IOM mapping, and channel mapping.

7.9.3.1 SCU Word Wrap

SCU word wrap provides data path checking in the SCU. The GPC sends a command word (data pattern) to the EMDM every 960 ms (1.04 Hz). Each time the pattern is altered slightly. The SCU compares the data received by the MIA with the response word sent through the SCU data path and back to the MIA. Figure 7-9 represents a block diagram of the SCU word process. If the SCU word wrap recognizes an incorrect match between the two data patterns, an immediate halt of the SCU will occur. This verification prevents three failures that are inherent to the current MDM. These failures are

A. Incorrect data sent to an IOM

B. Data sent to the wrong channel of an IOM

C. Data sent to the wrong IOM.

7.9.3.2 IOM Mapping

IOM mapping reduces the possibility of selecting the wrong IOM. The EMDM SCU/IOM interfaces logic, figure 7-10, has three redundant address latches. Each of these latches must supply the decoders and multiplexers with the same IOM address for successful transmittal or receiving of data. If the wrong IOM is selected during the transmittal of data, the data will not be transferred to the selected IOM output. During the receiving of data, the V-bit is set which will stop the further processing of the response word. For each case, BSR bits 3 and 4 will be set.

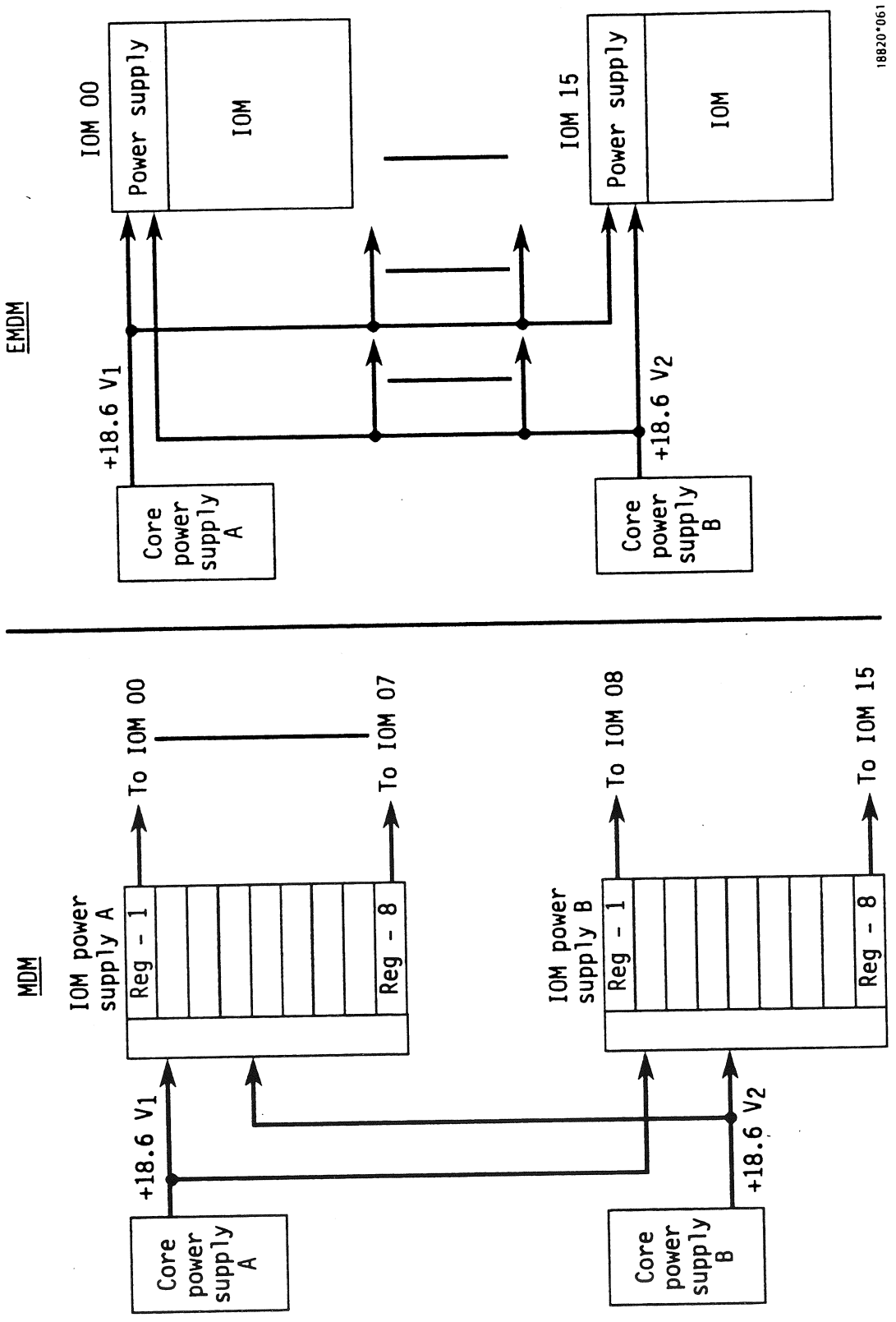


Figure 7-8.- MDM and EMDM power supply distribution.

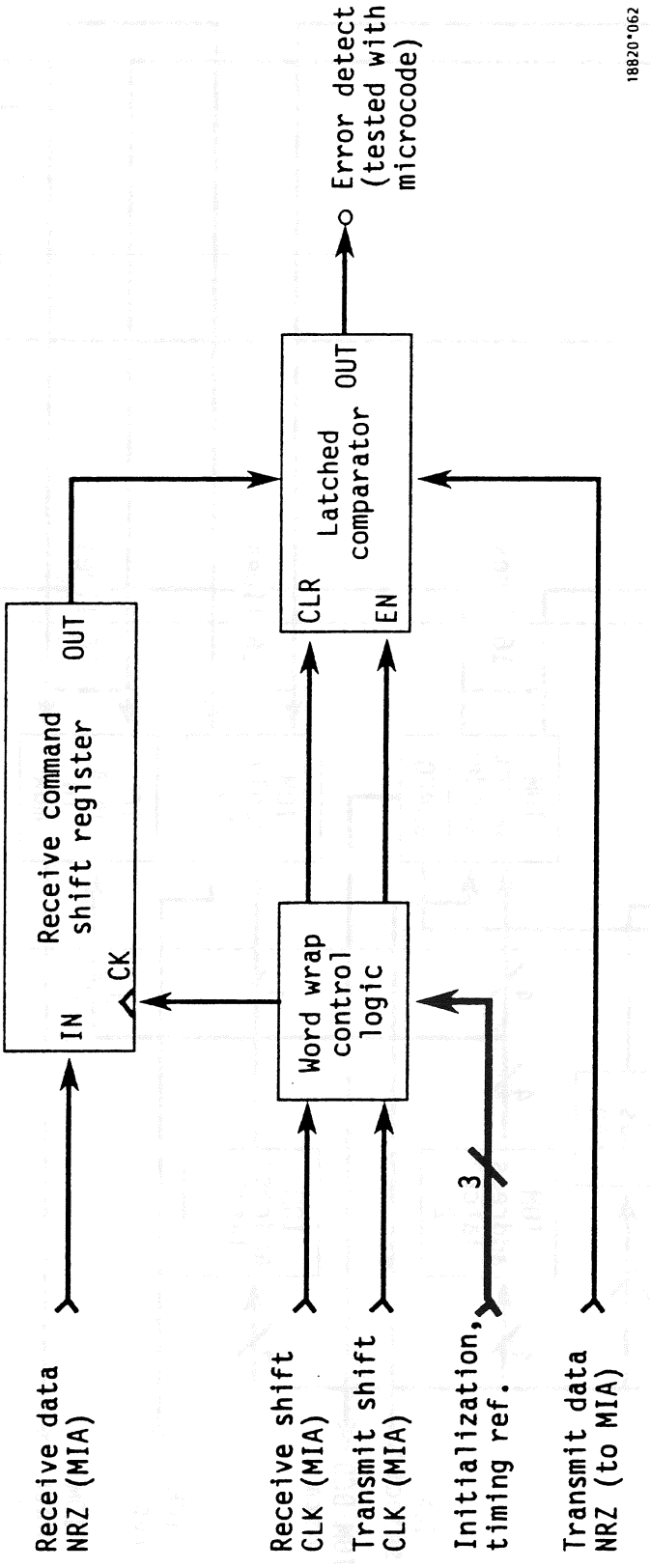


Figure 7-9.- SCU word wrap block diagram.

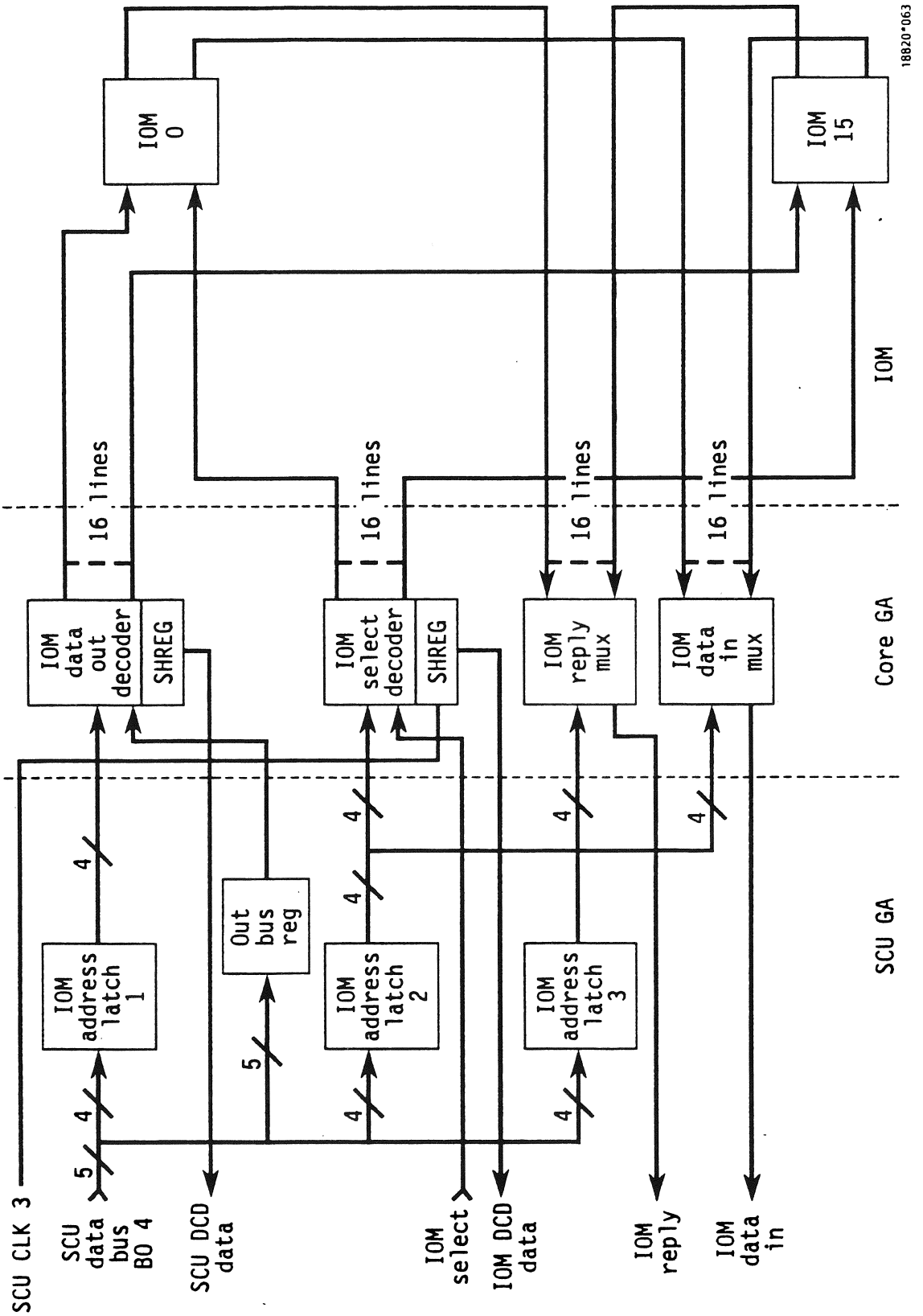


Figure 7-10.- EMDM SCU-to-IOM interface.

The IOM mapping does not protect for all IOM mapping failures. An example of an unprotected failure is with the analog input modules. If the wrong IOM address is sent to the A/D converter, the A/D could still select an analog bus from the wrong IOM. Another example of wrong IOM addressing is if there is a failure in addressing the sequence PROM.

7.9.3.3 Channel Mapping

Channel mapping is a fail-safe mechanism used in selecting the appropriate channel to be updated. The DOL and DOH IOM's are the only cards that implement this mapping scheme. As seen in the channel mapping block diagram, figure 7-11 the comparator must verify that both bus interface A and B went to update the same channel. If there is a conflict, then none of the channels will be updated, and an error message will be sent back to the SCU. This enhancement avoids the possibility of a channel being overwritten with eroneous data.

7.10 PHYSICAL CHARACTERISTICS

- Manufacturer - Sperry Flight Systems
  Phoenix, Arizona
  (MDM)

  Honeywell Satellite Systems Division
  Glendale, Arizona
  (EMDM)

- Part number - MC 615-0004-4X00 (MDM)
  MC 615-0004-7X00 (EMDM)

	Orbiter	SRB	Flexible
• Size (lxwxh-inches)	13.0x11.6x6.6	12.8x10.2x8.0	13.5x6.4x7.2
• Weight (≈pounds max.)	38.5	31.5	22.0
• Power consumption (watts)	~38-55 (MDM) ~41-51 (EMDM)	~53	~32
• Number of IOM's	16	8	8-field reconfigurable

Figure 7-12 is a photograph showing two views of a flexible MDM and one view of an orbiter MDM. The EMDM chassis is the same as the MDM's.

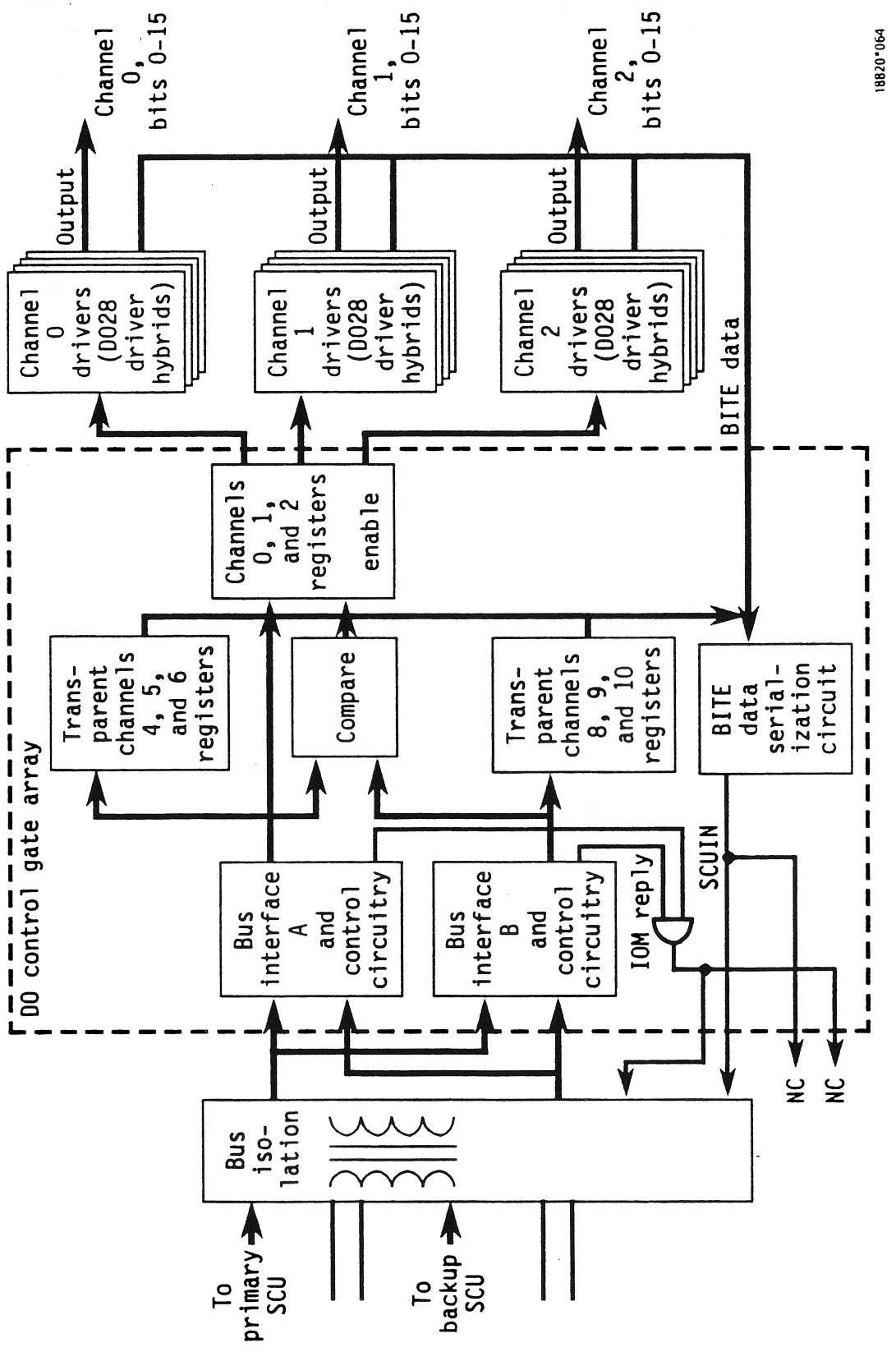
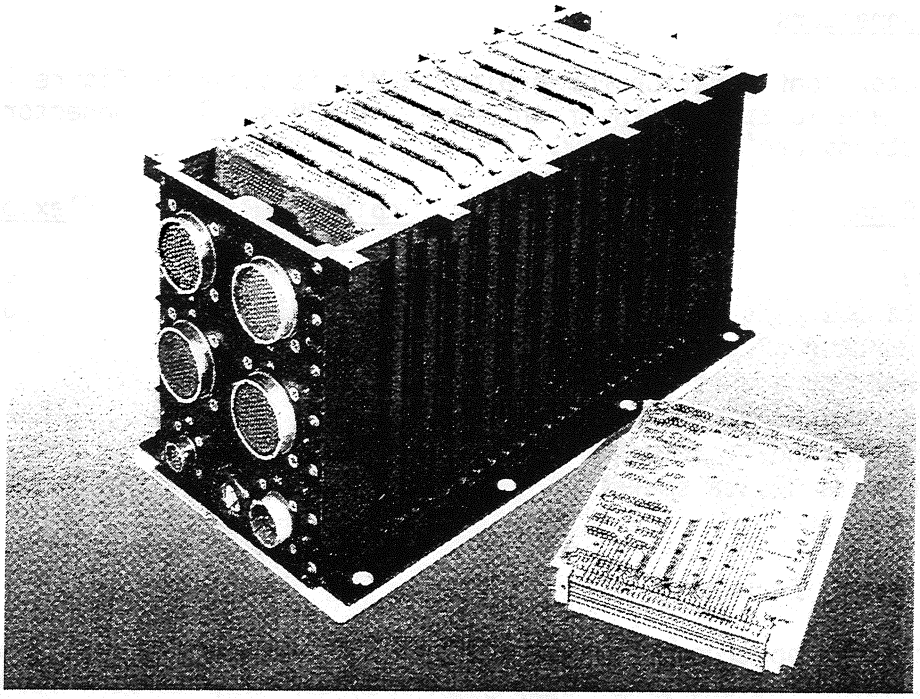
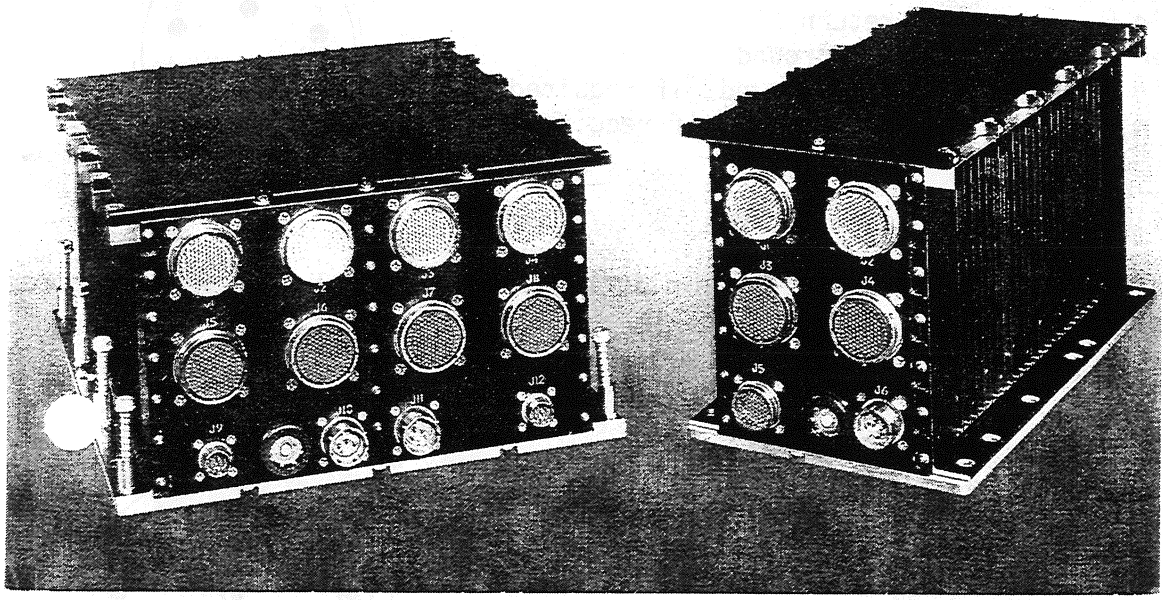


Figure 7-11.- Channel mapping block diagram.



(a) Flexible MDM with top and one card removed.



(b) Frontal view of dual-port orbiter MDM.

(c) Frontal view of flex MDM.

Figure 7-12.- Flexible and orbiter MDM's.

7.10.1 Connectors

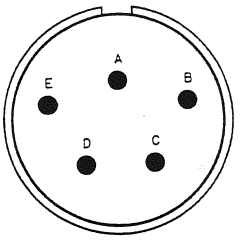
The connector configuration for orbiter MDM's is shown in figure 7-13. All connectors are located on the front panel of MDM's. The connectors and their functions are given below.

Connector	Orbiter MDM	Flexible MDM
Power plug	J10 and J11	J6
Serial data bus plug	J9 and J12	J5
Signal connector plug	J1,J2,J3, and J4	
Signal connector socket	J5,J6,J7, and J8	
Experiment command/data plug		J1 and J2
Experiment command/data socket		J3
Subsystem command/data socket		J4

7.10.1.1 Power Connectors

The orbiter MDM power supply 1 connector is J10 and the power supply 2 connector is J11. The one power supply in the FLEX is powered through connector J6. The pin assignment and a diagram of the pin configuration is given below.

Pin A	+28 V dc
Pin B	Dc return
Pin C	Signal ground
Pin D	Chassis ground, if required
Pin E	Shield ground, if required



3856. ART, 2

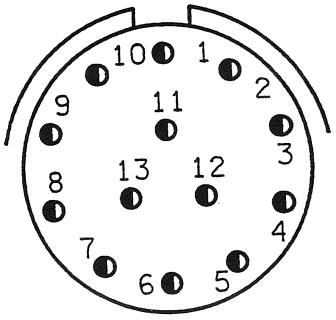
7.10.1.2 Data Bus Connectors

The pin assignment and connector diagram of the serial data bus connectors are given below.

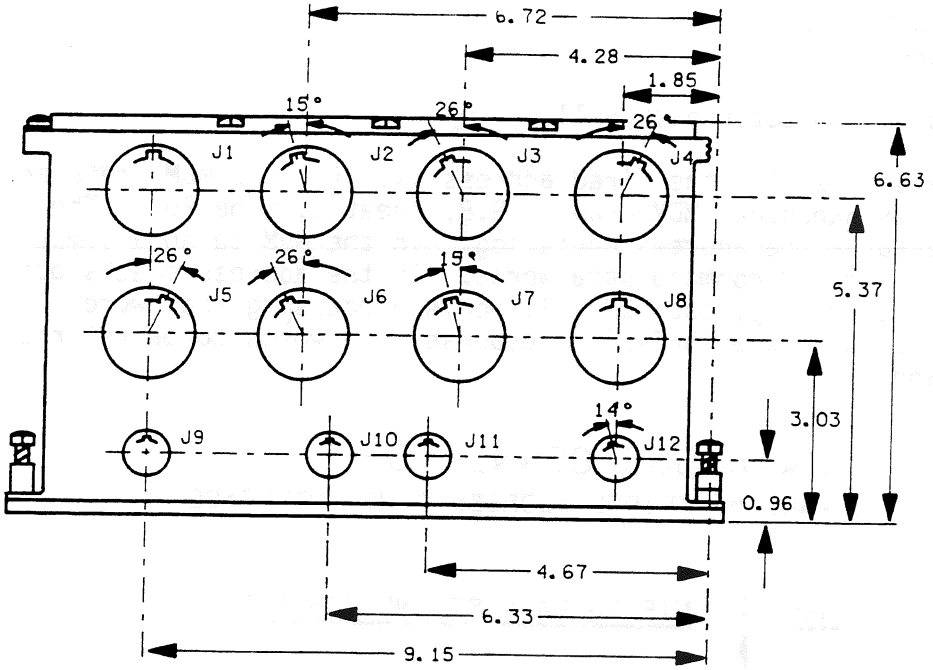
MDM Data Bus Connectors

J9 primary and J12 secondary

Pin 1	Logic 1
2	Spare
3	Logic 0
4	Spare
5	Bit 1 (MSB)
6	Bit 2
7	Bit 3
8	Bit 4
9	Bit 5 (LSB)
10	Spare



3856.Art 2



CONNECTORS			
MDM CONFIG	4X00 4X01 5X00 5X01	5X10 6XXX	
REF DES	PART NO.	PART NO.	FUNCTION
J1	NLSOE24-35P	MS27508E24F-35P	SIGNAL
J2	NLSOE24-35PA	MS27508E24F-35PA	SIGNAL
J3	NLSOE24-35PB	MS27508E24F-35PB	SIGNAL
J4	NLSOE24-35PC	MS27508E24F-35PC	SIGNAL
J5	NLSOE24-35SC	MS27508E24F-35SC	SIGNAL
J6	NLSOE24-35SB	MS27508E24F-35SB	SIGNAL
J7	NLSOE24-35SA	MS27508E24F-35SA	SIGNAL
J8	NLSOE24-35S	MS27508E24F-35S	SIGNAL
J9	NLSOE10-35P	MS27508E10F-35P	DATA BUS
J10	NBOH14-05PN	NBOH14-05PN	POWER
J11	NBOH14-05PW	NBOH14-05PW	POWER
J12	NLSOE10-35PA	MS27508E10F-35PA	DATA BUS

NOTES:
(UNLESS OTHERWISE SPECIFIED)

1. TOLERANCES:
   .XX = ±0.06
2. CONNECTOR MASTER KEYS ARE (REF) SHALL BE LOCATED AS SHOWN

*The EMDM has the same connector configurations.

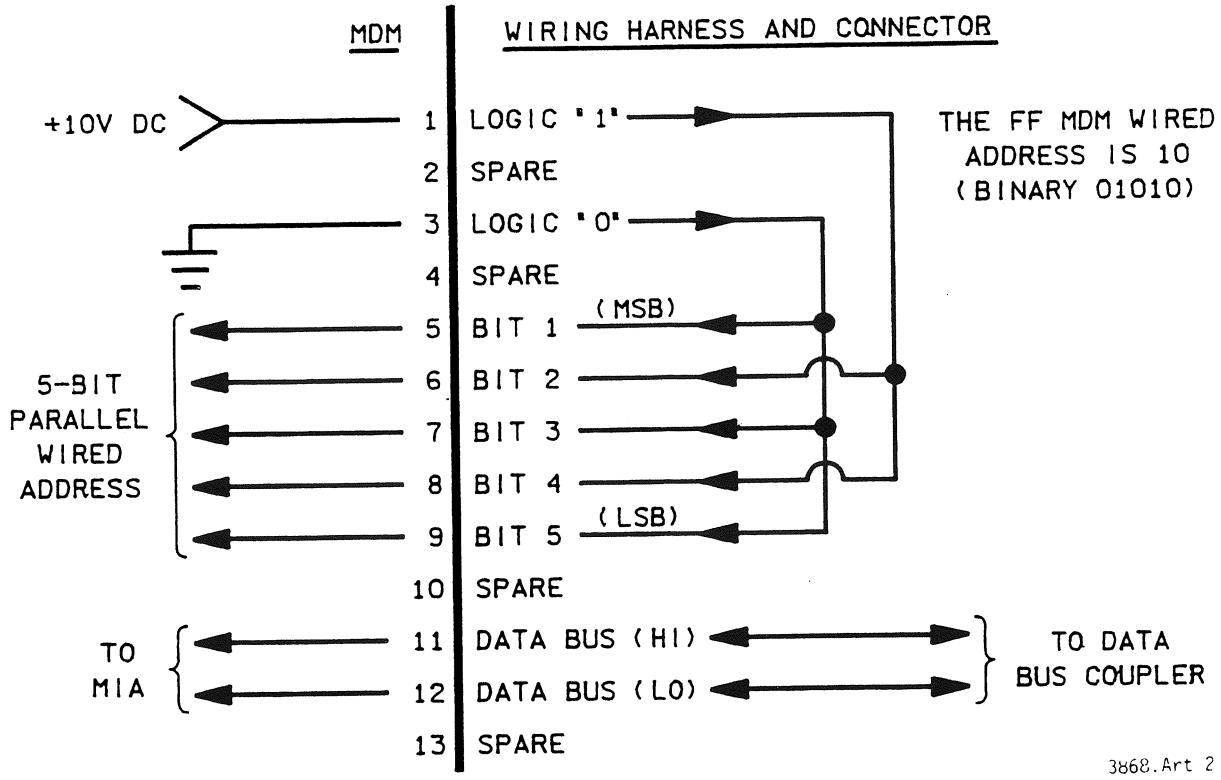
Figure 7-13.- MDM connector configuration.

11	Data bus (HI)
12	Data bus (LO)
Pin 13	Spare

Note: Shield tied to connector backshell.

The data bus connector is also the wired address plug for the MDM (reference Space Shuttle Systems Handbook, MDM drawing 8.8, sheet 1, zone P6). The wired address is used by the address check logic in the MDM to do a compare on all incoming command and command data words. If the address checks out and is synchronized correctly, the MDM will begin processing that word. The wired address is also used when building response data words to be returned to the MDM's commander.

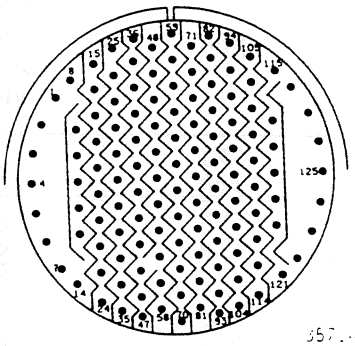
DATA BUS CONNECTOR
FLIGHT FORWARD MDM/WIRED ADDRESS PLUG INTERFACE



7.10.1.3 Signal Connectors

Signal connector pin assignments (128 pins/connector) may be found in the MDM Procurement Specification MC615-0004, Rev E, appendix III. The wiring between MDM connectors and IOM card connectors is identical for all MDM types.

MDM	Orbiter subsystems
Pin 1	
2	
3	
:	
127	IOM
128	unique



7.10.2 Controls

The power switches for flight critical forward, aft, and payload MDM's are located on PNL O6, figure 7-14.

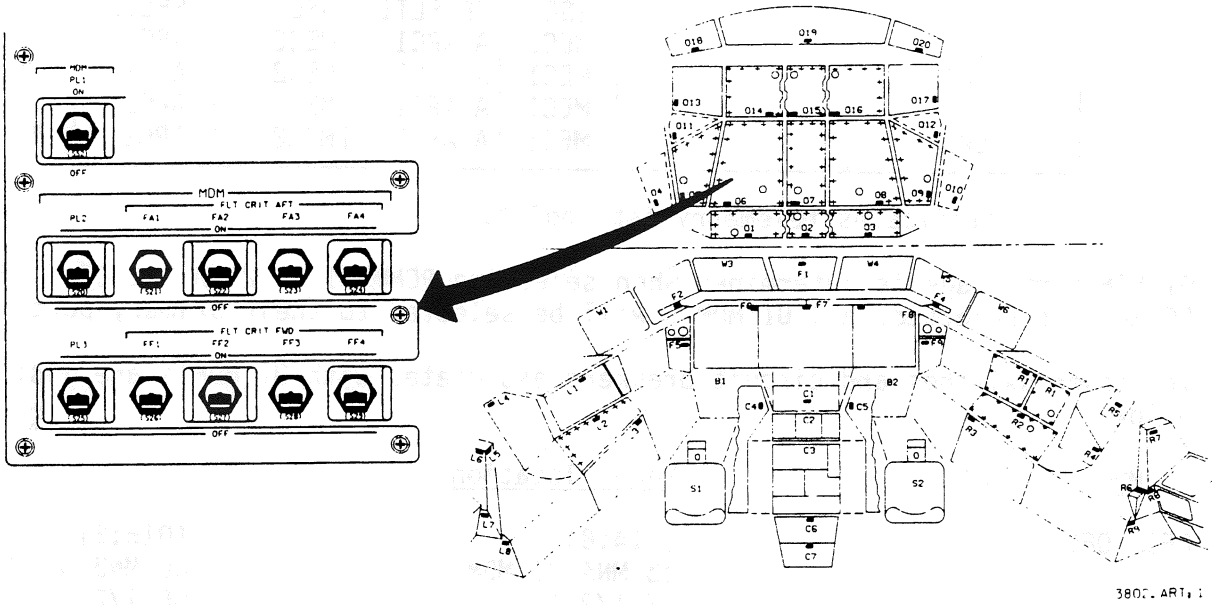


Figure 7-14.- MDM controls on PNL O6.

The PL3 switch does not have an MDM assigned to it at this time. The launch (LA, LF, and LM1) and SRB MDM's do not have power control switches available to the crew in the cockpit.

The panel switch provides control power for two RPC's that control the main bus power to each power supply in an MDM. Notice that telemetry pickoff points for MDM power status are on only one of the control buses for each MDM.

MDM	Panel 06 switch number	Power supply 1		Power supply 2	
		V_{CNTL}	$V_{MN:PCA}$	V_{CNTL}	$V_{MN:PCA}$
FF1	S26	AB2*	A FPC1	AB3	B FPC2
FF2	S27	BC2*	B FPC2	BC3	C FPC3
FF3	S28	CA3*	C FPC3	CA2	A FPC1
FF4	S29	BC1*	B FPC2	BC3	C FPC3
FA1	S21	AB1*	A APC4	AB2	B APC5
FA2	S22	BC1*	B APC5	BC2	C APC6
FA3	S23	CA2*	C APC6	CA1	A APC4
FA4	S24	CA3*	C APC6	CA1	A APC4
PF1	S52	AB1	A FPC1	AB3*	B FPC2
PF2	S20	BC1*	B FPC2	BC3	C FPC3
LF1		GSE	PREFLT1	GSE	PREFLT2
LA1		GSE	PREFLT1	GSE	PREFLT2
LM1		GSE	PREFLT1	GSE	PREFLT2
LL1		MEC1	A APC1	MEC2	B APC2
LL2		MEC1	A APC1	MEC2	B APC2
LR1		MEC1	A APC1	MEC2	B APC2
LR2		MEC1	A APC1	MEC2	B APC2

*Indicates telemetry data point.

OI MDM port usage is determined when selecting PCMMU 1 or 2 on PNL C3. If PCMMU 1 is selected, all OI MDM's will be selected to their primary port.

The power switches and circuit breakers associated with OI MDM's are listed below.

MDM	Panel location	
OF1, OF2	(O14:B) cb MNA OI MDM OF 1/2 A	(O15:B) cb MNB OI MDM OF 1/2 B
OF3, OF4	(O14:B) cb MNA OI MDM OF 3/4 A	(O16:B) cb MNC OI MDM OF 3/4 B
OA1	(O17:D) MDM OA 1/2/3	MNA, MNB
OA2	(O17:D) MDM OA 1/2/3	MNB, MNC
OA3	(O17:D) MDM OA 1/2/3	MNA, MNC

Flight critical and payload MDM's can be port moded by the crew using the DPS UTILITY display or by the ground using DEU equivalent commands to the DPS UTILITY display.

The ground interface provides both power and control interfaces to the LA1, LF1, and LM1 MDM's.

7.10.3 Cooling and Locations

With active cooling, the MDM's have a minimum operating and nonoperating temperature of -65° F, a maximum operating temperature of 160° F, and a maximum nonoperating temperature of 165° F. Locations of the orbiter MDM's are given in figures 7-15 through 7-20. The MDM, its location, and type of cooling interface are listed below.

MDM	Location	Cooling
FF1	Forward AV bay 1	Coldplate/water
FF2	Forward AV bay 2	Coldplate/water
FF3	Forward AV bay 3	Coldplate/water
FF4	Forward AV bay 2	Coldplate/water
FA1	Aft AV bay 4	Coldplate/Freon
FA2	Aft AV bay 5	Coldplate/Freon
FA3	Aft AV bay 6	Coldplate/Freon
FA4	Aft AV bay 6	Coldplate/Freon
PF1	Forward AV bay 1	Coldplate/water
PF2	Forward AV bay 2	Coldplate/water
OF1	Forward AV bay 1	Coldplate/water
OF2	Forward AV bay 2	Coldplate/water
OF3	Forward AV bay 3A	Coldplate/water
OF4	Flight deck	Coldplate/water
OA1	Aft AV bay 4	Coldplate/Freon
OA2	Aft AV bay 5	Coldplate/Freon
OA3	Aft AV bay 6	Coldplate/Freon
LF1	Forward AV bay 1	Coldplate/water
LA1	Aft AV bay 6	Coldplate/Freon
LM1	Midfuselage	Coldplate/Freon
LL1	Left SRB forward integrated electronics assembly	Ambient air/passive
LL2	Left SRB aft integrated electronics assembly	Ambient air/passive
LR1	Right SRB forward integrated electronics assembly	Ambient air/passive
LR2	Right SRB aft integrated electronics assembly	Ambient air/passive
Flexible	Payload bay pallets	Coldplate/radiators

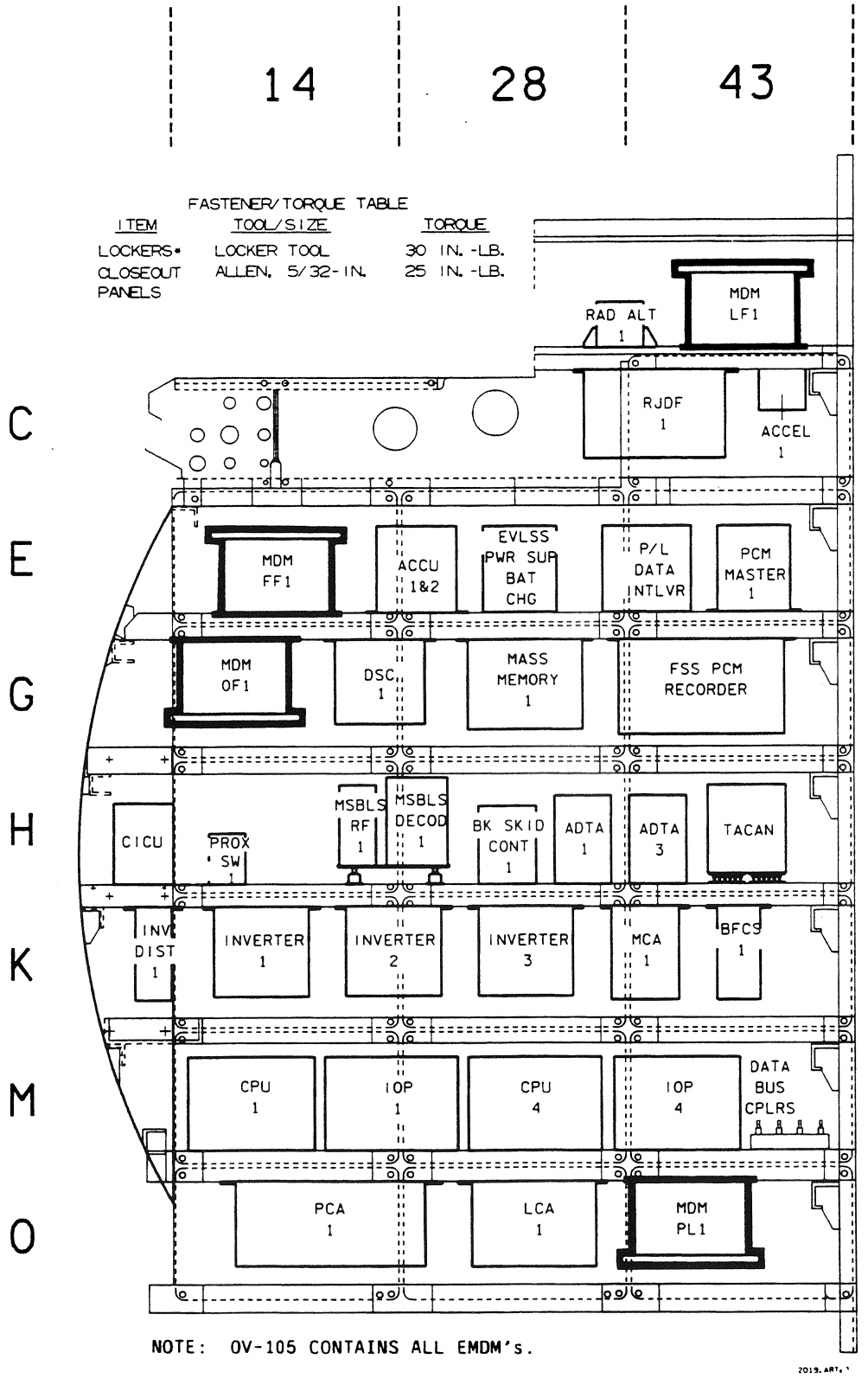


Figure 7-15.- Orbiter MDM locations, AV bay 1.

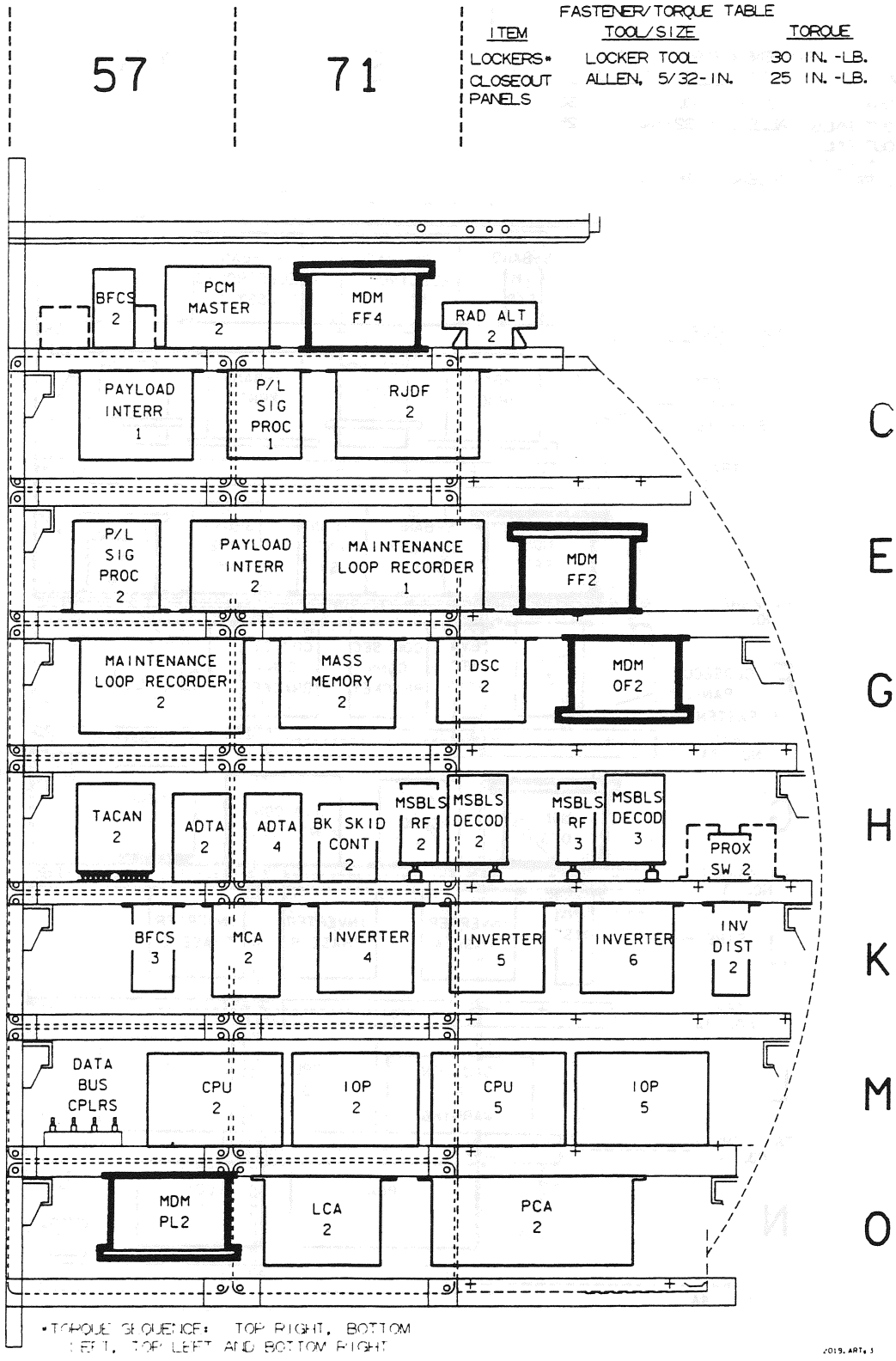


Figure 7-16.- Orbiter MDM locations, AV bay 2.

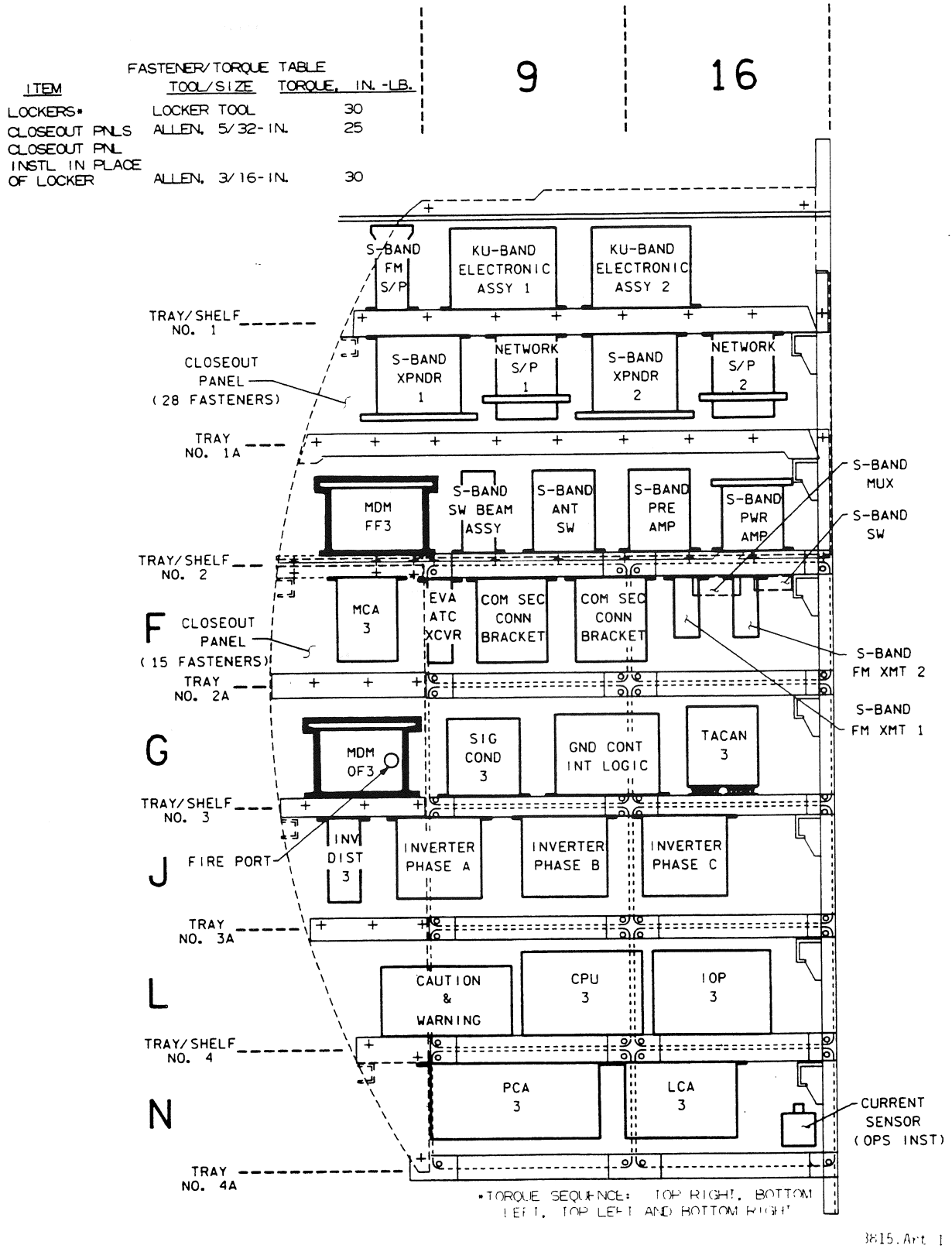


Figure 7-17.- Orbiter MDM locations, AV bay 3A.

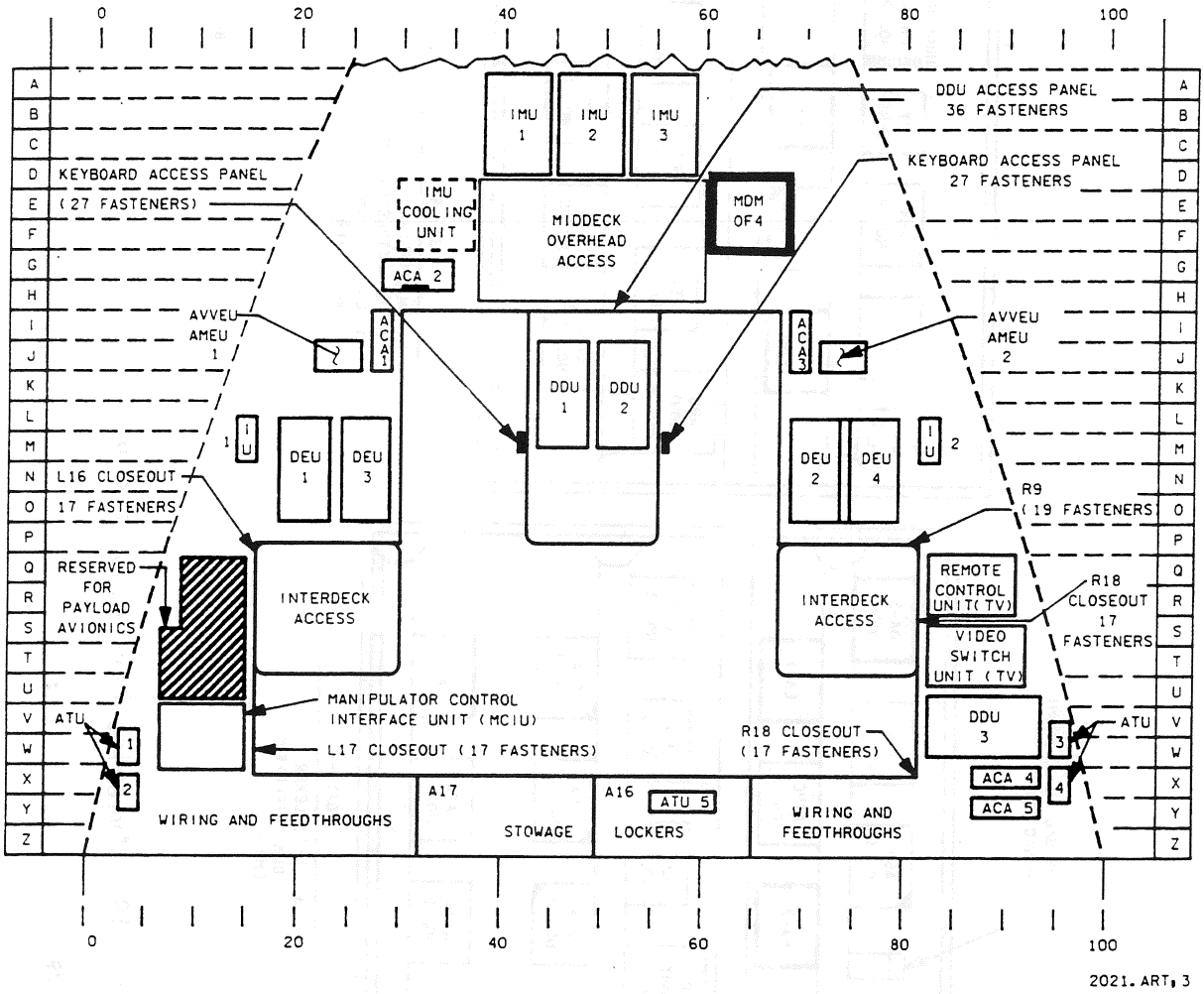


Figure 7-18.- Orbiter MDM location, flight deck.

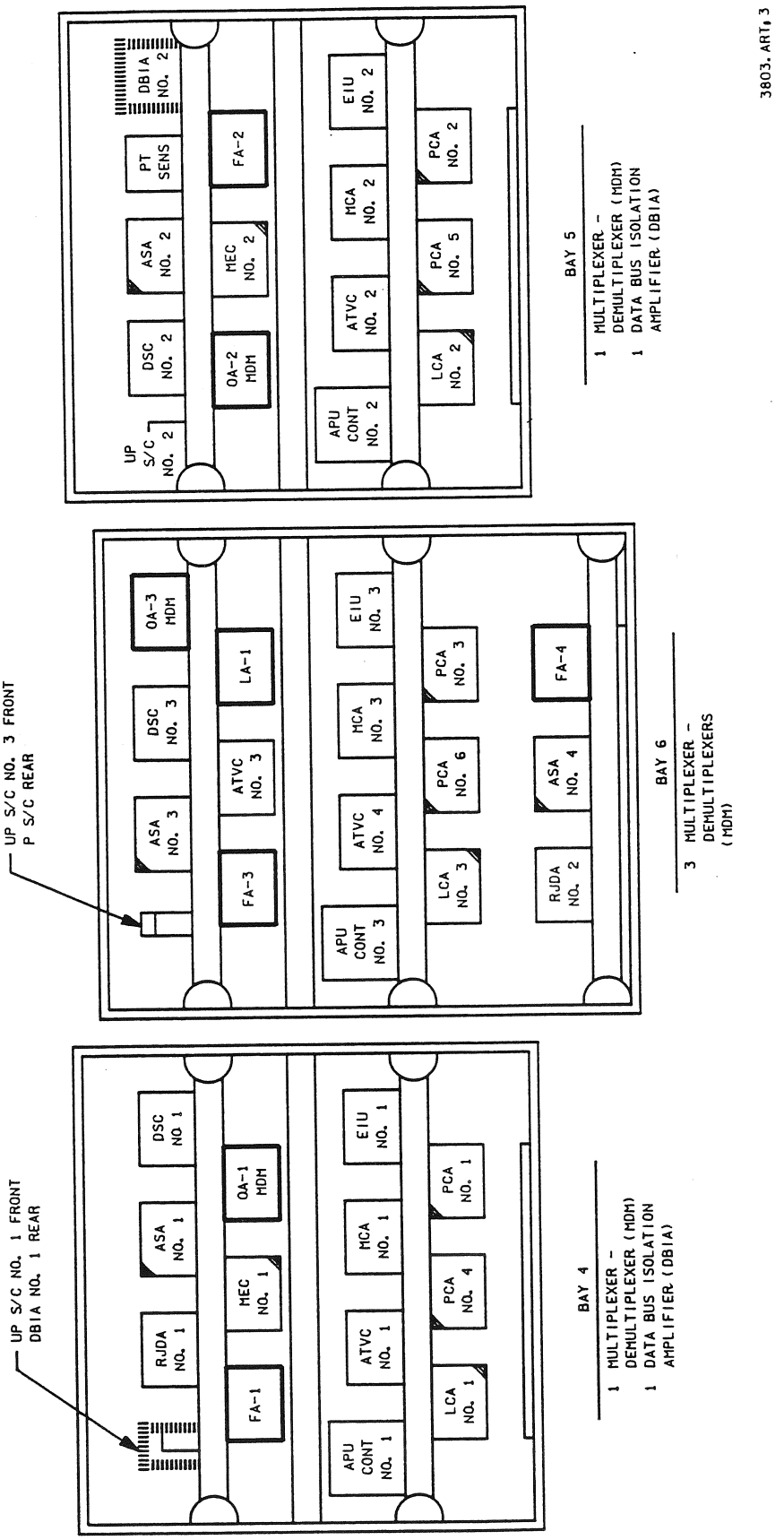


Figure 7-19.- Avionics bays 4, 5, and 6.

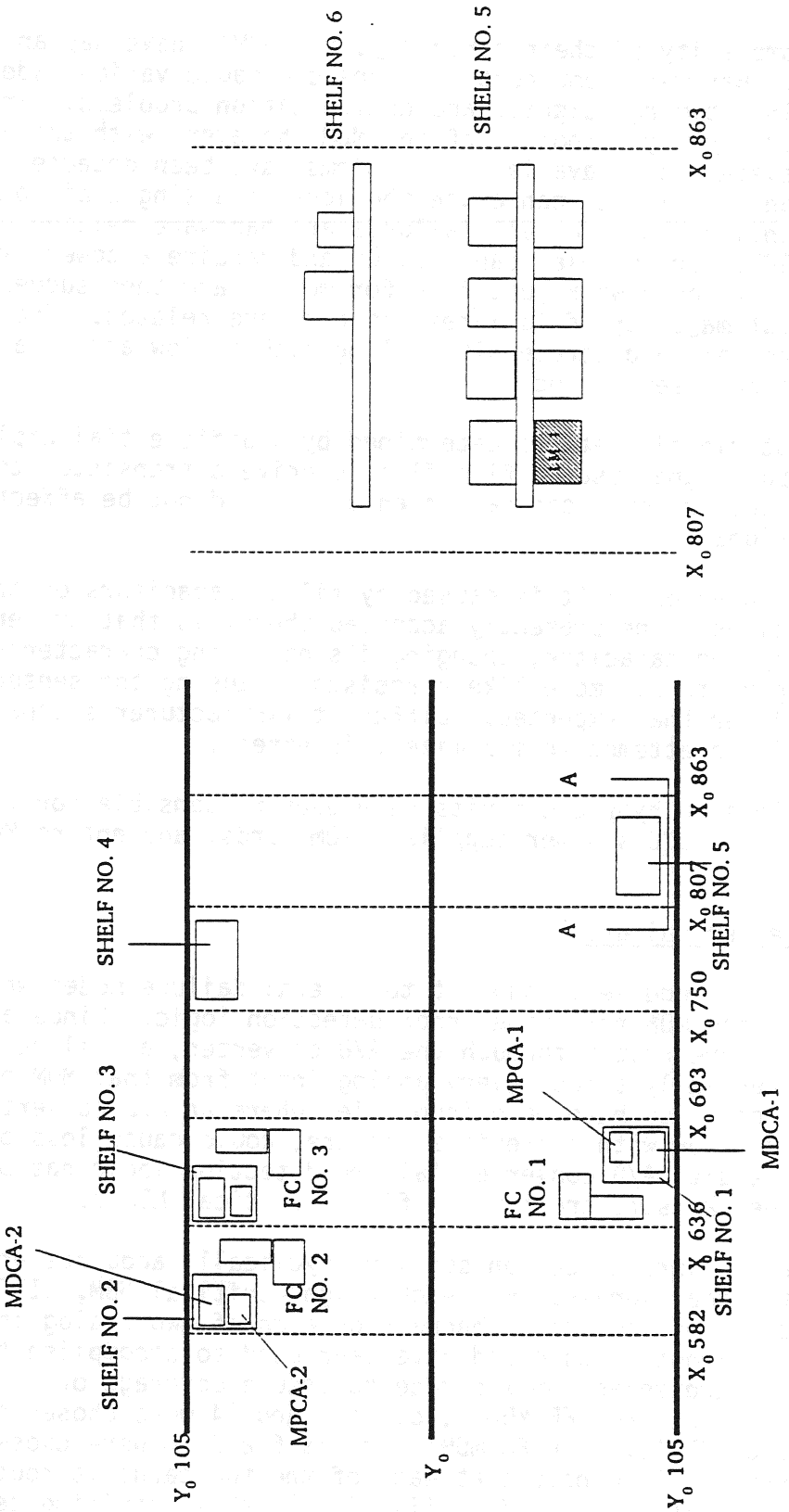


Figure 7-20.- Midfuselage (EDO).

7.11 FAILURE MODES

Because of the complexity of their circuitry, the MDM's have had an assortment of failures. Hardware component failures can cause various addressing logic, data routing, timing, signal, and communication problems. Problems can occur in virtually any component of the MDM; however, with the BITE testing and the systems data available, problems have been detected and corrected preflight. Failures can cause the loss of a single bit of data to the loss of the entire MDM. Not all failures are hardware related or explainable. The SCU, for example, can lock up and require a power cycle to get going again. It could work just fine for months and then suddenly lock up again. The vast majority of failures are hardware related. The most common two failures are a discrete bit failing high or low and an analog input signal that has been biased.

The discrete input signal state is determined by a differential amplifier. The discrete output signal uses a flip-flop to drive a transistor to its on or off state. Either circuit can fail high or low and not be affected by an input or output signal.

The analog signal bias or shift is caused by filter capacitors on the input lines changing values. The presently accepted theory is that ambient humidity is absorbed by the capacitor, changing its operating characteristics. The capacitor begins to act more like a resistor, causing the sensed signal to be higher or lower than expected. Different manufacturer's capacitors have been tried in an attempt to minimize this effect.

Failures of portions in hybrid circuits have been responsible for the functional loss of MIA's, SCU's power supplies, IOM cards, and entire MDM's.

7.11.1 A/D Converter Failure

The MDM A/D converter module is subject to several failure modes which are not detectable by the MDM real-time error detection logic. Since all analog input measurements are routed through the A/D converter, a failure in the converter can potentially affect every analog input from that MDM port. Certain failure scenarios have been identified where an A/D converter failure, in conjunction with a previous failure, could cause loss of vehicle and crew. As a result, A/D converter failure detection logic has been implemented in the GPC software for all flight-critical MDM's.

The A/D converter failure detection software cyclically acquires and tests selected MDM reference signals. For each flight-critical MDM, BITE reads are performed for two consecutive channels on each of two analog input cards. Any analog input cards could have been used to accomplish this; however, specific card selection was made to assure coverage of "critical" analog input interface. For FF MDM's, cards 1 and 14 were chosen due to RHC's, SBTC's, and RPTA's. For FA MDM's, cards 6 and 14 were chosen due to SRB chamber pressure. Also note that each of the two cards is routed through a different differential MUX (fig. 7-21), thus providing detection for failures in either MUX.

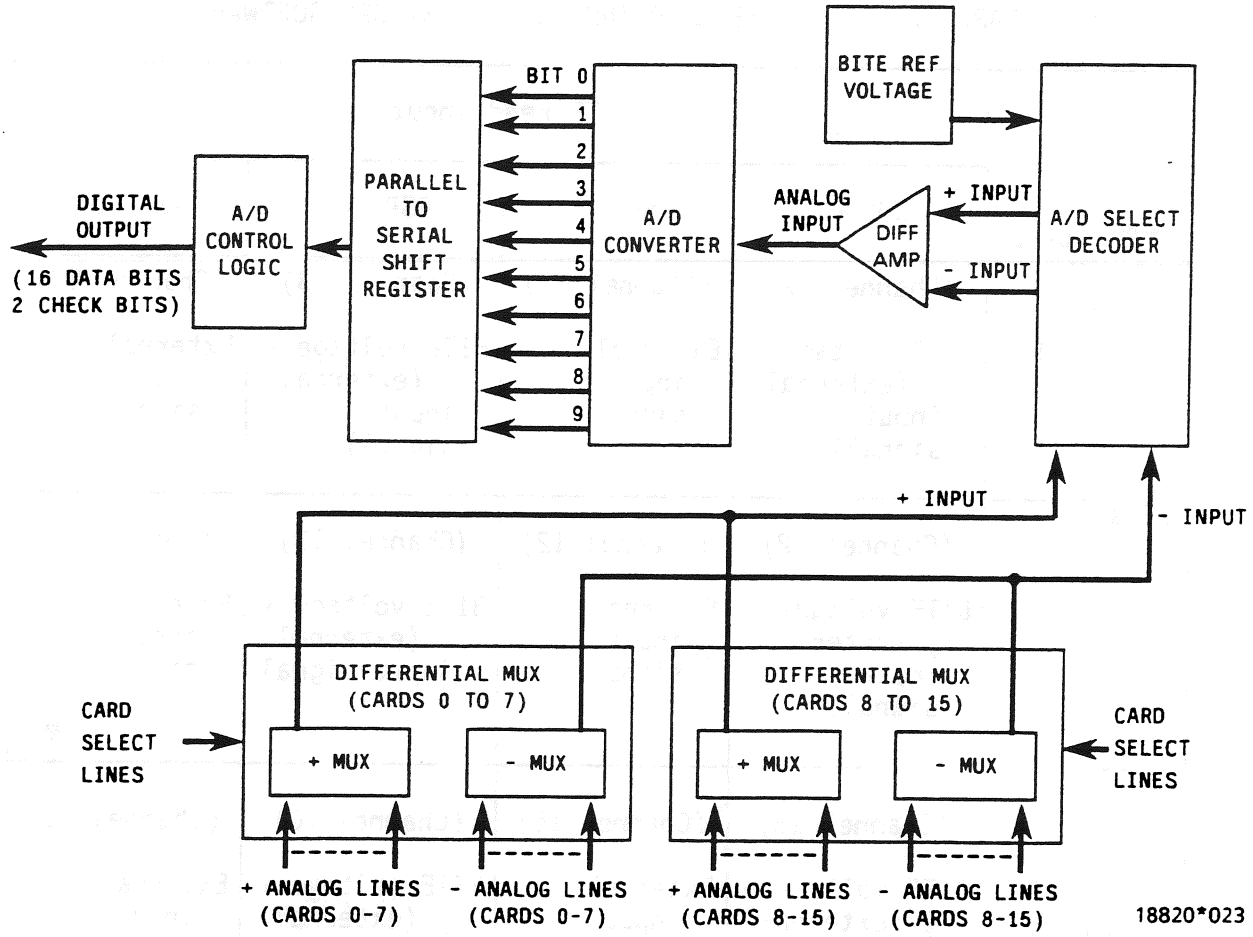


Figure 7-21.- A/D converter module.

There are four response data words for each of the two cards on an MDM. These four words consist of two response words from each of two consecutive channels. The first response words is the BITE voltage plus half of the external input signal. The second response word is the external input signal itself. Table 7-III summarizes the BITE read inputs used by the GPC software to detect A/D failures:

TABLE 7-III.- BITE READ INPUTS USED BY GPC SOFTWARE

MDM	CARD	BITE read inputs			
		RSP WD1	RSP WD2	RSP WD3	RSP WD4
FF's	1	(Channel 12) BITE voltage + ½ (external input signal)	(Channel 12) External input signal	(Channel 13) BITE voltage + ½ (external input signal)	(Channel 13) External input signal
	14	(Channel 12) BITE voltage + ½ (external input signal)	(Channel 12) External input signal	(Channel 13) BITE voltage + ½ (external input signal	(Channel 13) External input signal
FA's	6	(Channel 15) BITE voltage + ½ (external input signal)	(Channel 15) External input signal	(Channel 16) BITE voltage + ½ (external input signal	(Channel 16) External input signal
	14	(Channel 15) BITE voltage + ½ (external input signal	(Channel 15) External input signal	(Channel 16) BITE voltage + ½ (external input signal	(Channel 16) External input signal

The GPC software will perform the indicated computations for each I/O card BITE read data set:

BITE_POS_2V = [(RSP_WD1) - 1/2(RSP_WD2)]
BITE_NEG_2V = [(RSP_WD3) - 1/2(RSP_WD4)]

These computed BITE values are then compared against the reference value. The comparison is made only to the most significant five bits:

REF_POS_2V = 00110XXXXXXXXXX
REF_NEG_2V = 11001XXXXXXXXXX

If both of the computed BITE values fail to compare with their corresponding reference voltages, the software will annunciate BCE string A and B messages for FF MDM's or BCE string C and D messages for FA MDM's. The MFE and HFE PROM sequences will be bypassed.

The I/O error condition is accomplished by modifying the bus commander BCE program via utilization of illegal BCE instructions. This approach permits use of standard I/O error fault detection logic to generate the appropriate BCE bypasses across both the PASS and BFS. Currently, the BFS will downmode the entire string that has the faulty A/D converter. A CR has been written to require the BFS to bypass the appropriate PROM sequences, and NOT downmode the string.

An MDM A/D BITE FAIL STATUS flagword is downlisted and available to the DPS operator. If either of the BITE values (POS_2V or NEG_2V) fails to compare with its reference voltage, the corresponding bit in the flagword will be set. If a problem is detected on one card but not the other, the appropriate bit will be set in the flagword, but no I/O error condition will result.

The A/D converter could fail such that all analog inputs on the MDM are affected. In this case, the A/D converter fail detection logic would detect a "miscompare" for both cards. All analog inputs on that MDM will be bypassed, as evidenced by the BCE STRG messages. The following indications would be seen on console:

A. On MSK 1301

1. Two BCE STRG messages would be annunciated. (strings A and B for FF MDM's; strings C and D for FA MDM's.)

2. The I/O error log would contain the BCE for the corresponding MDM, the ID/EL for the corresponding PROM sequence, and a status word of 0000 0004. The status word indicates an illegal BCE instruction.

B. On MSK 1302

3. The MDM A/D BITE fail status word will have bits set for the corresponding MDM. (This word is available only in high data rate.)

4. The PASS will bypass the PROM sequences (A and B, or C and D) for the corresponding MDM. This can also be seen on the DDD's.

5. If the BFS is up, it will stop tracking the corresponding string (also seen on DDD's).

C. On MSK 1364

6. The MDM A/D BITE fail status word is broken down by bit. The word FAIL should appear next to both cards for the MDM that had the A/D problem.

What does it mean if a single bit is set in the A/D BITE status word; i.e., only one card shows the fail indication? In this case, no I/O error condition would result and the data would not be bypassed. The failure could be on that particular card, causing bad data to be input from that card. Or the failure could be in one of the differential MUX blocks (fig. 7-16) in the A/D converter. This type of failure would affect only half of the analog input cards in that MDM. The other half of the inputs go through the other differential MUX.

7.11.2 SRB MDM Power Supply Failure

In 1986, a dual failure scenario associated with the SRB MDM power supply was identified that was potentially catastrophic. This failure scenario resulted in the loss of output of both power sources in the SRB MDM for at least a brief period of time, causing discrete outputs to be reset, resulting in a loss of thrust vector control (TVC) for that SRB. The identification of this scenario led to investigations that identified a single-point failure that would cause the same result. Due to the single-point failure, the SRB MDM was modified to include a "power separation" capability. Power separation is discussed more fully in section 7.6.3.

7.11.3 MDM Power Cycling

The failure recovery procedures for MDM's often includes power-cycling the affected MDM. MDM power cycling exhibits both detrimental and beneficial effects on the vehicle systems, depending upon the flight activities and specific LRU's which are channelized through the affected MDM's. The basic philosophy on MDM power cycling is:

> For GPC-related failures, flight-critical MDM's will not immediately be power-cycled unless there is a specific requirement for the power cycle. The only identified requirement for immediate MDM power cycles following GPC failures is for cleanup of OMS TVC enable commands.

These effects of power cycling are summarized as follows:

A. IMU

The IMU operate discretes are channelized through FF 1, 2, and 3 and are nominally in the ON or "1" state. A loss of the operate discrete for 2 to 4 seconds causes the IMU to cage, following which appropriate recovery procedures are required to regain use of the IMU. These recovery procedures are not available during dynamic flight. If a GPC commanding FF 1, 2, or 3 fails to software halt or if it has a flight critical 1, 2, or 3 transmitter failure, all discretes in the affected MDM, including the operate discrete, will remain in their last commanded state. If the affected MDM is power-cycled, all discretes will be reset to 0 or OFF and will remain in that state because there is no GPC available to refresh the discretes until the MDM is regained via restringing. Thus, the operate discrete will be lost, and the IMU will

cage within 2 to 4 seconds. Therefore, unless there are specific requirements to do so (see OMS discussion below), it is inappropriate to power cycle FF 1, 2, or 3 following GPC-related failures because the affected IMU will be lost. If the power cycle is not performed, the IMU will be available following restringing or BFS engage and will potentially be usable depending upon how far it has drifted while it was not being commanded. A point should be made with respect to how quickly an MDM must be power-cycled to avoid caging an IMU. If there is no GPC available to refresh the operate discrete when the MDM is power-cycled as is the case for GPC failures, it does not really matter how quickly the MDM is cycled. The IMU will always cage because the operate discrete will be OFF until the MDM is regained via restringing or BFS engage. However, if the MDM is being actively commanded by a GPC, the operate discrete will immediately be refreshed to an ON state following the power cycle (output commands do not require an I/O RESET). In this case, the IMU would not cage unless the crewmember allowed the MDM to be OFF for greater than 2 to 4 seconds.

B. OMS

When there are multiple MDM or OMS hardware failures, the potential exists for inadvertently shutting off an OMS engine when an FA MDM is power-cycled. Therefore, it is not appropriate to power-cycle an FA MDM during a burn. However, the OMS TVC enable discretes are channelized through the FF MDM's; and, for specific GPC failures, the FF MDM's must be power-cycled. The primary TVC enable discretes are channelized through FF 1 and 4 for the left and right OMS engines, respectively; the secondary TVC enable discretes are channelized through FF 2 and 3 for left and right OMS engines, respectively. If strings 1 or 4 are lost due to a GPC failure, the secondary gimbals are selected manually by the crew. However, the primary enables may still be set in the affected FF MDM resulting in both the primary and secondary gimbals enabled at the OMS engine. With both sets of gimbals enabled, the affected OMS gimbals will lock up. A power cycle of the affected FF MDM (1 or 4) is required to remove the primary enable for the TVC gimbals so that the secondary gimbals can function properly. Unique software processing during ascent can result in erroneous automatic secondary gimbal selection for certain types of GPC failures. A power cycle of the affected FF MDM (2 or 3) is again required to prevent gimbal lockup.

C. MPS

During nominal entries and aborts, the aft vehicle compartment ET umbilical cavity and OMS pods are purged with helium to ensure that there is no residual fuel from a possible fuel leak. This purge is automatically initiated at a relative velocity of 4500 fps and reduces the potential for fire hazard posttouchdown. There are two purge valves plumbed in series such that, if either valve loses the open command, the purge will stop. The open commands for the two valves are channelized through FA 3 and 4. Following a GPC fail to halt or flight-critical transmitter failure affecting either FA 3 or 4, a power cycle of FA 3 and 4 will result in a loss of the open command to one of the two purge

valves and the purge will be terminated. Therefore, it is inappropriate to cycle FA 3 or 4 below a relative velocity of 4500 fps following GPC-related failures. However, there are circumstances under which cycling MDM power is beneficial to the MPS. During ascent through the completion of the MPS dump sequence, certain MPS solenoid valves are commanded on. Following the completion of the dump sequence, all MPS solenoid valves are commanded off. If a GPC fails between lift-off and the completion of the MPS dump, there is the potential of hanging up the MPS solenoid valves in the ON or energized state for the duration of the flight. This would not be an operational impact to the flight, but would reduce significantly the lifetime of the solenoid. There are two ways to procedurally avoid this problem. The first is a power cycle of the affected MDM prior to the OPS 2 transition which would reset the solenoid output commands and turn off the valve. The second solution is to restring prior to the OPS 2 transition to allow a software cleanup of the solenoids. We have chosen the latter solution for procedural implementation. For GPC failures on an RTLS, TAL or AOA abort, MDM power cycling may be performed following the transition to OPS 9 for MPS valve problems. DPS will coordinate with Booster to determine if the power cycle is required.

D. Vent doors

Initially it was thought that GPC failures during the vent door closure at TIG-25 minutes could result in motor close discrete output commands being hung up at the affected MDM. However, further investigation revealed that following the vent door sequence, the software leaves the cyclic outputs for the vent doors in a state which protects for GPC single point failures and lightning (i.e., the B reset command remains on). When the string is recovered via restringing, the affected MDM outputs for vent door command will then be driven to a state which ensures there will be no phase-to-phase shorts on ac buses when the doors are opened during entry. If a GPC fails at the vent door opening sequence during entries (nominal entry, RTLS, TAL, or AOA), the motor open discrete output command can get hung up and remain hung up when the string is recovered at the OPS 9 transition. However, the TCS sequence for commanding the doors to the purge position first resets all open and close commands, thereby avoiding any potential problems.

E. Crew displays

Many of the cockpit indicator lights are driven via discrete outputs through the FF MDM's. A GPC failure could result in a light being inappropriately stuck on. The subject lights include some of the C&W matrix lights, the DAP lights, eyebrow panel lights, etc. For the subject failure mode, the lights could be reset with an MDM power cycle. However, discussions with the STS-6 crew indicate that stuck-on lights do not create sufficient confusion to justify MDM power cycling.

valves and the purge will be terminated. Therefore, it is inappropriate to cycle FA 3 or 4 below a relative velocity of 4500 fps following GPC-related failures. However, there are circumstances under which cycling MDM power is beneficial to the MPS. During ascent through the completion of the MPS dump sequence, certain MPS solenoid valves are commanded on. Following the completion of the dump sequence, all MPS solenoid valves are commanded off. If a GPC fails between lift-off and the completion of the MPS dump, there is the potential of hanging up the MPS solenoid valves in the ON or energized state for the duration of the flight. This would not be an operational impact to the flight, but would reduce significantly the lifetime of the solenoid. There are two ways to procedurally avoid this problem. The first is a power cycle of the affected MDM prior to the OPS 2 transition which would reset the solenoid output commands and turn off the valve. The second solution is to restring prior to the OPS 2 transition to allow a software cleanup of the solenoids. We have chosen the latter solution for procedural implementation. For GPC failures on an RTLS, TAL or AOA abort, MDM power cycling may be performed following the transition to OPS 9 for MPS valve problems. DPS will coordinate with Booster to determine if the power cycle is required.

D. Vent doors

Initially it was thought that GPC failures during the vent door closure at TIG-25 minutes could result in motor close discrete output commands being hung up at the affected MDM. However, further investigation revealed that following the vent door sequence, the software leaves the cyclic outputs for the vent doors in a state which protects for GPC single point failures and lightning (i.e., the B reset command remains on). When the string is recovered via restringing, the affected MDM outputs for vent door command will then be driven to a state which ensures there will be no phase-to-phase shorts on ac buses when the doors are opened during entry. If a GPC fails at the vent door opening sequence during entries (nominal entry, RTLS, TAL, or AOA), the motor open discrete output command can get hung up and remain hung up when the string is recovered at the OPS 9 transition. However, the TCS sequence for commanding the doors to the purge position first resets all open and close commands, thereby avoiding any potential problems.

E. Crew displays

Many of the cockpit indicator lights are driven via discrete outputs through the FF MDM's. A GPC failure could result in a light being inappropriately stuck on. The subject lights include some of the C&W matrix lights, the DAP lights, eyebrow panel lights, etc. For the subject failure mode, the lights could be reset with an MDM power cycle. However, discussions with the STS-6 crew indicate that stuck-on lights do not create sufficient confusion to justify MDM power cycling.

7.11.4 EMDM Serial Input/Output (SIO) Power Supply Failure

A. Background

The EMDM serial input/output (SIO) module power supply exceeds the current limit of the current limiting circuit when the box is powered up with only one bus. This failure results from the increase in the capacitive load on the IOM power supply to support serial digital transmission. Honeywell (vendor) has seen that, upon the SIO module powerup, the current exceeds the limiting circuit if the 28 V input rise time is between 6-18 msec. This overcurrent condition turns off the SIO power supply regulator drive transistor and holds it in an off condition until the EMDM is power cycled.

B. Signature

SIO card 3 ① - BCE STRG X IMU
MTU
STKR ②

SIO card 11 - BCE STRG X ADTA ③
MLS ③
NSP ②

① FF4 does not use SIO card 3
② Message will only be annunciated with FF1 and FF3
③ Message only applicable for G1/G3/G8/G9

NOTE: In G2, the KU-RADAR/RNDZ RADAR (ID/EL - 15/45) will get bypassed if FF3 SIO card 3 fails. However, no corresponding fault message will be annunciated.

C. Recovery

Power cycle the EMDM and do an I/O reset to regain GPC/MDM communication.

7.11.5 EMDM S/N 221 SIO Card 11

During the EMDM transparency tests performed in the JAEL (May 1990), a condition was discovered where the EMDM outputs would be reset following an input power bus transient on one of the two power buses. The EMDMs were retrofitted by Honeywell. However, during a review of the build records it was determined that EMDM S/N 221 SIO card 11 was not fixed. This particular card holds the MLS BITE TEST CMD and ADTA and MLS inputs. The MLS BITE TEST CMD output had no impact because it is used only for FCS checkout and the command could always be re-sent if a power transient occurred. The ADTA and MLS inputs should not be affected by transients. However, if the power transients are intermittent enough to interrupt the inputs, an I/O reset should regain communication with the LRU. Currently, this EMDM resides in OV-105 FF2 slot (November 1991).

BACKUP FLIGHT
CONTROLLER


BACKUP FLIGHT
CONTROLLER

SECTION 8
BFC AND PASS/BFS INTERFACES

CONTENTS

Section		Page
8.1	BFC OVERVIEW	8-1
8.1.1	Features/Performance	8-1
8.2	BFC HARDWARE	8-3
8.2.1	BFC CRT Select Logic	8-9
8.2.2	GPC HALT Relay	8-9
8.2.3	Power Up/Down Logic	8-12
8.2.3.1	+28 V dc Detector Circuit	8-12
8.2.3.2	+5 V dc Detector Circuit	8-12
8.2.3.3	Power-Off Reset Circuit	8-15
8.2.4	GPC Select Logic and Delay Filter	8-15
8.2.5	Hand Control Engage Driver (HCED)	8-18
8.2.6	Engage/Disengage Logic	8-21
8.3	PASS/BFS INTERFACE OVERVIEW	8-25
8.3.1	Disengaged/Engaged Manual Ops Transition/ Initialization	8-27
8.3.1.1	OPS 0 Processing	8-29
8.4	BFS AND PASS MCDS INTERACTIVE PROCESSING	8-30
8.4.1	General Characteristics	8-30
8.4.2	GPCIPL DEU Characteristics	8-31
8.4.3	Keyboard Processing	8-32
8.4.4	DEU BITE	8-32
8.4.5	SPEC Handling Nuances	8-32
8.5	GPC HALT OPERATIONS	8-32
8.6	GPC SELECT OPERATIONS (GPC OUTPUT SWITCH)	8-33
8.7	ENGAGE OPERATIONS	8-36
8.7.1	PASS Response to BFS Engage	8-36
8.7.2	Hand Controller Engage Driver (HCED) Operations	8-38
8.7.3	BFS Engage Software and Mechanisms	8-38
8.7.4	Control Bus Down Impacts	8-40
8.7.5	DDU +15 V dc Power Loss Impacts	8-41
8.7.6	RPCA Power Loss Impacts	8-41
8.8	DISENGAGE OPERATIONS	8-41
8.8.1	BFS Software Response to a Disengage	8-41
8.8.2	PASS Disengage Software	8-42
8.8.3	Control Bus Down Impacts	8-42
8.9	BFS/PASS I/O ERROR PROCESSING	8-44
8.10	BFS/PASS GPC DATA BUS CONTROL	8-44
8.11	LISTEN COMMAND PROCESSING	8-46
8.12	HIP POCKET BFS-RELATED IFM PROCEDURES AND PERIPHERAL DATA	8-49
8.13	REFERENCES	8-58

SECTION 8
BFC AND PASS/BFS INTERFACES

8.1 BFC OVERVIEW

There are three functions the Backup Flight Controller (BFC) performs. The first, and most important, is to provide manual switchover capability from the PASS to the BFS. Second, the BFS CRT selection logic is provided for manipulating CRTs between the PASS and BFS computers. Lastly, the GPC MODE switch HALT signal is routed through a relay in the BFC module. There are three BFC units on the Orbiter, each with two separate BFC modules. Because the modules within the BFC are independent from each other (not redundant to each other), we will consider them as separate units unless otherwise noted. Figure 8-1 presents an overview of the BFC module interfacing components and power sources.

8.1.1 Features/Performance[6]

Manufacturer: Rockwell International Autonetics Group

Part number: MC615-0023

Drawing number: 15180-507

Description: Each BFC unit contains two identical modules, A and B. Each module has BFS CRT select logic, Hand Controller Engage Driver (HCED) logic, engage/disengage logic, and a GPC mode halt signal relay.

Orbiter location: BFC 1 - Avionics bay 1
BFC 2 - Avionics bay 2 (fig. 8-2)
BFC 3 - Avionics bay 2

Power:

BFC	DDU	Control bus	GPC
1A	L DDU A	CNTL AB3	1
1B	L DDU B	CNTL AB3	4
2A	L DDU C	CNTL AB3	2
2B	R DDU A	CNTL CA1	5
3A	R DDU B	CNTL CA1	3
3B	R DDU C	CNTL CA1	N/A

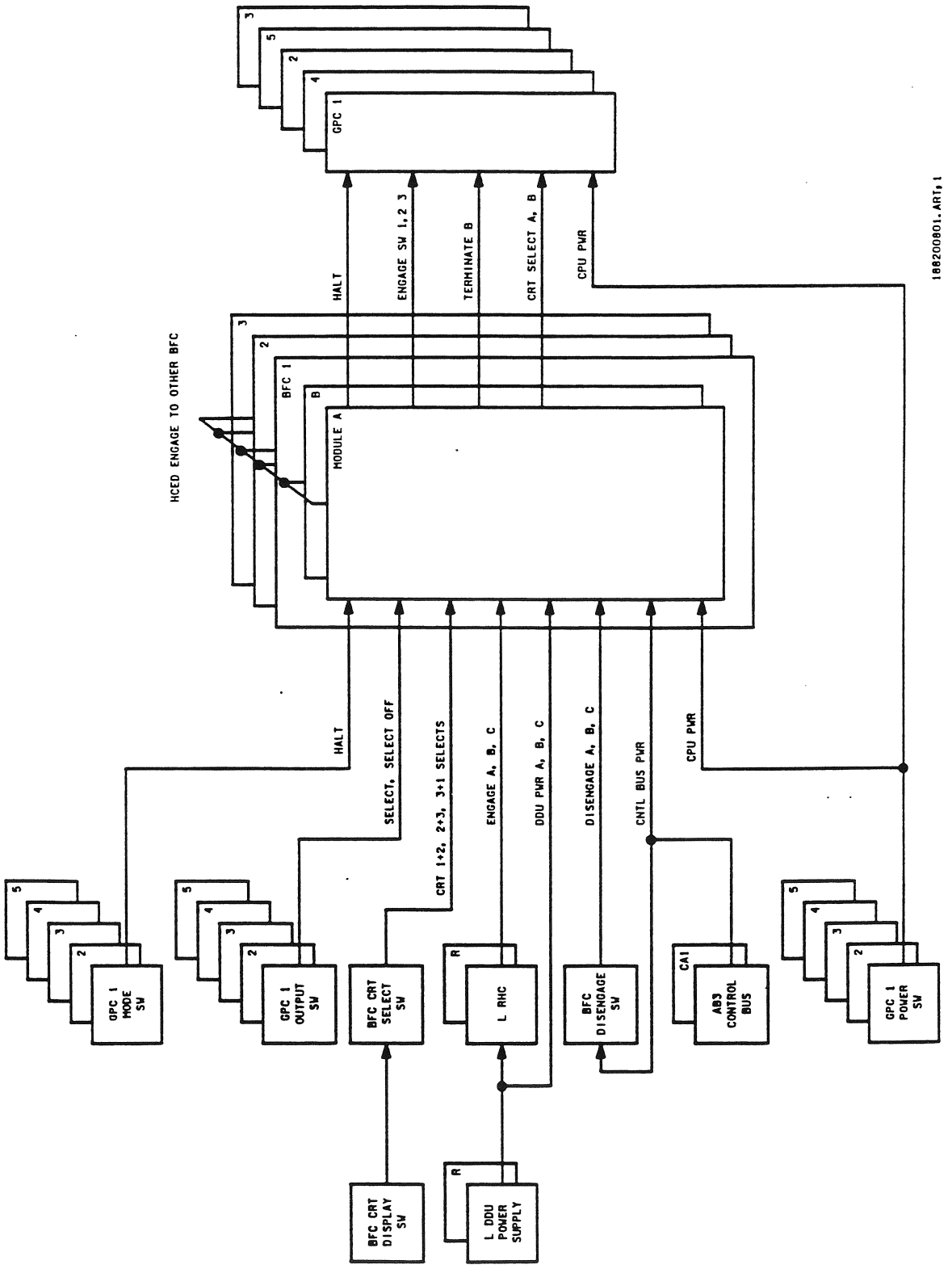


Figure 8-1.- BFC module interface overview.

Power consumption: 7.3 watts per module

Size: Rectangular box 15.0 by 3.5 by 7.0 inches (fig. 8-3)

Weight: 12.5 lb

Environment: Temperature - Minimum, 0° F
Maximum, 140° F

Pressure - Minimum, 8 psia for 165 minutes
Maximum, 18 psia

8.2 BFC HARDWARE[6]

As stated before, each BFC box contains two identical modules of electronics, modules A and B. Each module is independent from the other and can be thought of as separate pieces of equipment. Only five BFC modules are used on the Shuttle. The sixth module (BFC 3B) is not used except for the HCED and engage signal path from the HCED to the other BFC modules.

Before we discuss the BFC itself, we need to first take a quick look at the physical interface to the outside world. Figure 8-2 shows the location of the BFC in avionics bay 1 and 2. Figure 8-3 give the dimensions of the box and the relative locations of the pin connectors. Tables 8-I, II, and III give the pin layout of the connectors.

Now, let's look at the individual modules. The BFC is made up of six major functional areas:

A. BFS CRT select logic

B. GPC HALT signal relay

C. Power up/down logic

D. GPC select logic and delay filter

E. HCED

F. Engage/disengage logic

The functional blocks and signal interfaces are highlighted in figure 8-4. The reader should note that the engage/disengage, power up/down logic, and GPC select logic are shown in one major block in figure 8-4 because they are very much related when talking about the overall function of the BFC to effect an engage. Some may prefer to think of all three modules as one "engage/disengage function."

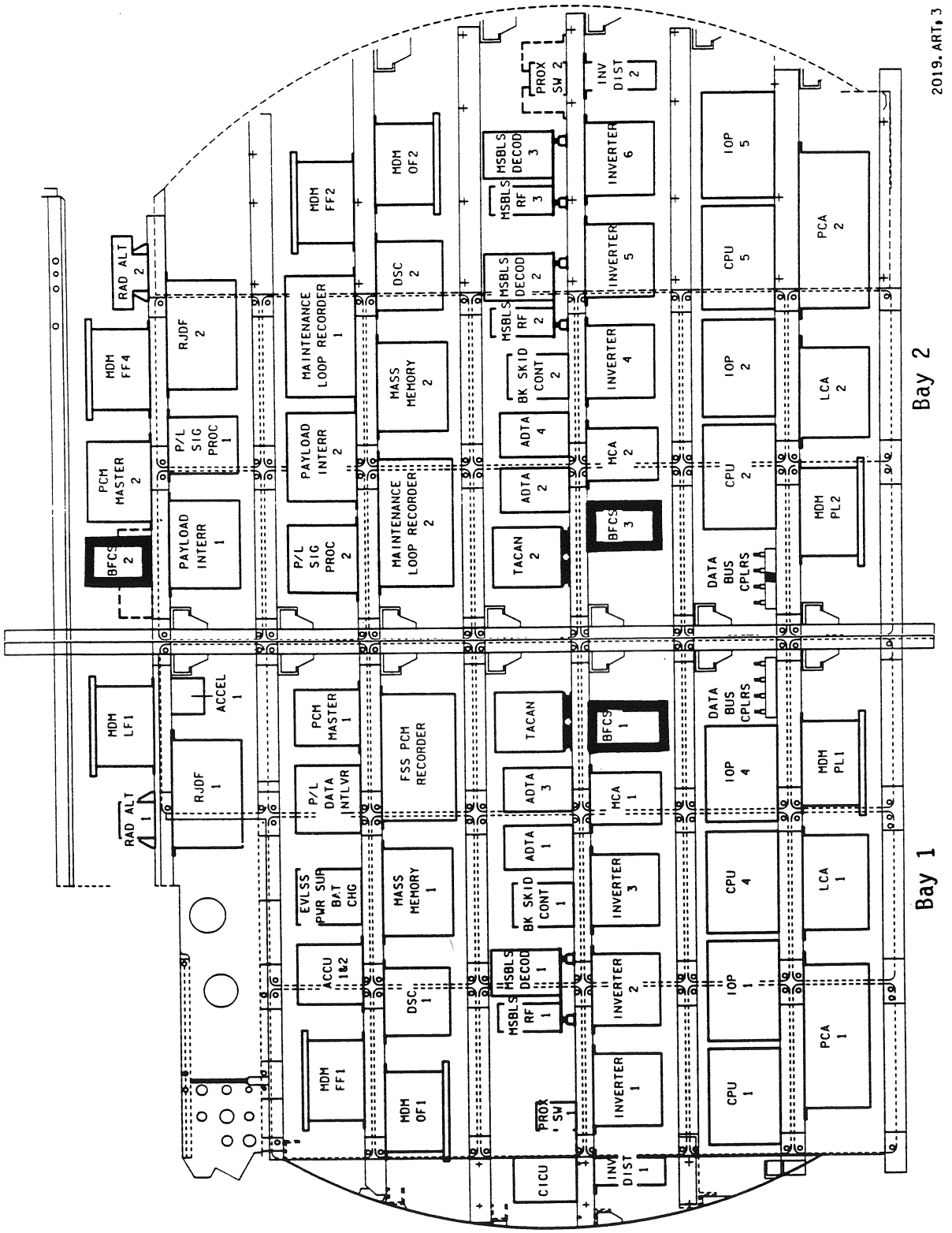


Figure 8-2.- Avionics bays 1 and 2.

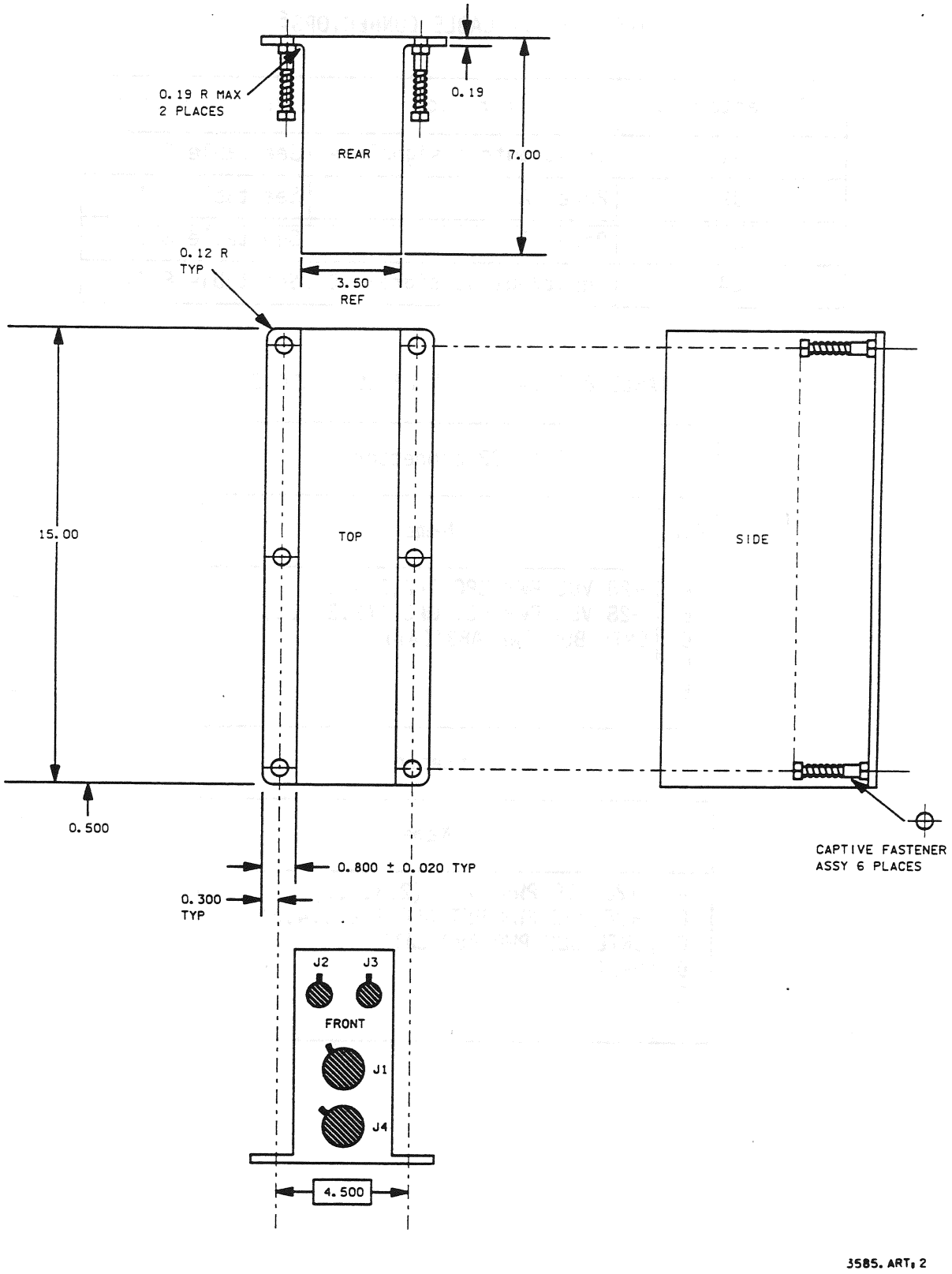


Figure 8-3.- BFC dimensions[5].

TABLE 8-I.- CABLE CONNECTORS[6]

Connector ID	Function	Pin description
J1	Input/output signals A	See table 8-III
J2	Power A1	See table 8-II
J3	Power B1	See table 8-II
J4	Input/output signals B	See table 8-III

TABLE 8-II.- J2 AND J3 CONNECTORS

(a) J2 connector	
Pin	Name
A	+28 VDC PWR GPC 1(2,3,4,5)
B	+28 VDC PWR RET GPC 1(2,3,4,5)
C	CNTL BUS PWR AB3(CA1)
D	HALT
E	
F	

J3 connector	
Pin	Name
A	+28 VDC PWR GPC 1(2,3,4,5)
B	+28 VDC PWR RET GPC 1(2,3,4,5)
C	CNTL BUS PWR AB3(CA1)
D	HALT
E	
F	

TABLE 8-III.- J1 AND J4 CONNECTORS[6]

Pin	Name	Pin	Name
1	CNTL BUS PWR	27	T ENGAGE SW 3
2	L ENGAGE A	28	C ENGAGE SW 3
3	L ENGAGE B	29	T BFC CRT SELECT A
4	L ENGAGE C	30	C BFC CRT SELECT A
5	CRT 1 + 2	31	
6	R ENGAGE A	32	T TERMINATE B
7	R ENGAGE B	33	C TERMINATE B
8	R ENGAGE C	34	
9	CRT 2 + 3	35	
10	BFC DISENGAGE A	36	
11	BFC DISENGAGE B	37	
12	BFC DISENGAGE C	38	T BFC CRT SELECT B
13		39	C BFC CRT SELECT B
14	SELECT	40	L(R) RHC HCED SIGNAL A(B,C)
15		41	
16	CRT 3 + 1	42	
17	SELECT OFF	43	T HALT
18	L(R) RHC ENGAGE A(B,C)	44	C HALT
19	L(R) RHC ENGAGE A(B,C) RET	45	
20	L(R) DDU + 15 VDC PWR A(B,C)	46	
21		47	
22	GND SIGNAL RETURN	48	
23	T ENGAGE SW 1	49	
24	C ENGAGE SW 1	50	
25	T ENGAGE SW 2	51	
26	C ENGAGE SW 2	52	

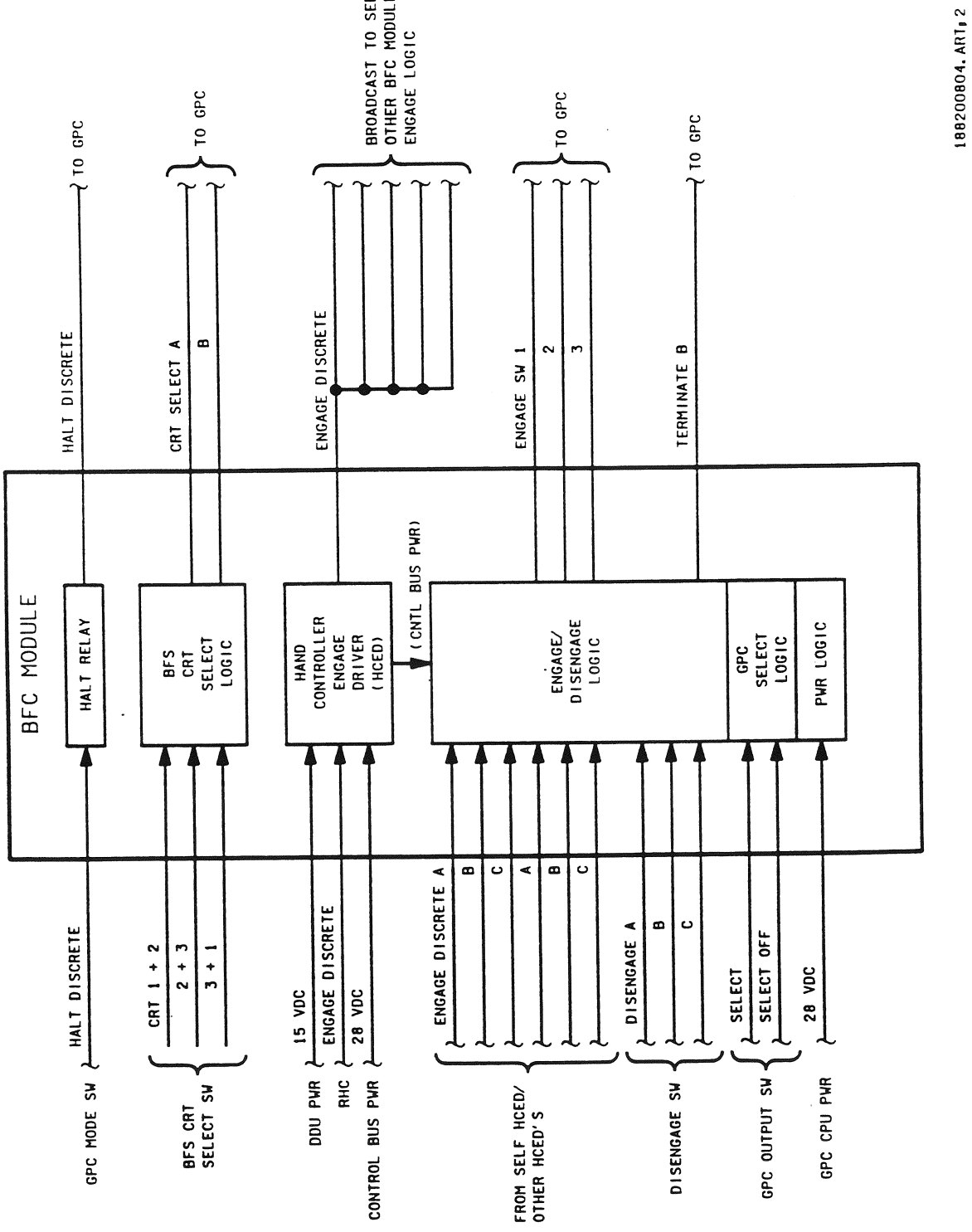


Figure 8-4.- BFC module functional overview.

8.2.1 BFC CRT Select Logic

The function of the BFC CRT select logic is to receive the three discrete signals from the BFC CRT SELECT switch, convert the signals to binary information, and broadcast the CRT pair selected for BFS GPC control to the interfacing GPC.

As shown in figure 8-5, the BFC CRT SEL 1, 2, and 3 signals from the BFC CRT SELECT switch are filtered and electrically isolated upon entering the BFC. The filter and optical isolator keep extraneous line noises from affecting the CRT selection as well as isolating potential problems in one BFC module from the other BFC modules. The three discrete signals are then combined into a two-digit, binary information signal (four possible combinations). Each digit is then split into a true/compliment (T/C) pair for transmission to the interfacing GPC. The T/C pair arrangement is done to reduce the effects of picking up spurious signals/interference between the BFC and GPC. The power for the CRT select logic comes from the +5 V power supply in the BFC.

8.2.2 GPC HALT Relay

The function of the halt relay is to reduce the possibility of electromagnetic interference on the halt signal by providing a break in the wire length from the GPC MODE switch (HALT contact) to the interfacing GPC. As shown in figure 8-6, the halt signal (single wire) from the interfacing GPC MODE switch is sent to the BFC halt relay coil. This +28 V activates the relay which provides a ground (negative logic 1 signal) for the true side of the T/C pair of wires which then goes to the halt discrete receiver in the interfacing GPC. When no signal is present (the relay is de-energized), the relay contact provides a ground to the complement side of the T/C wire pair and leaves the true side floating with respect to ground.

Note that, except for the physical location, the halt relay is totally independent from the rest of the BFC. Its location in the BFC is only a matter of convenience in the layout of the Shuttle internal wiring.

NOTE

This relay warrants a closer look by the reader because it shows the function of all discrete transmitters in the Shuttle DPS and the relation of the T/C pair. We normally think of a transmitter as providing a positive signal to a receiver and, typically, this sort of relationship is shown in the functional drawings. For the design of the Shuttle, this is not really the case. For this system, a positive signal from the transmitter is really "closing" the signal path from the receiver to ground.

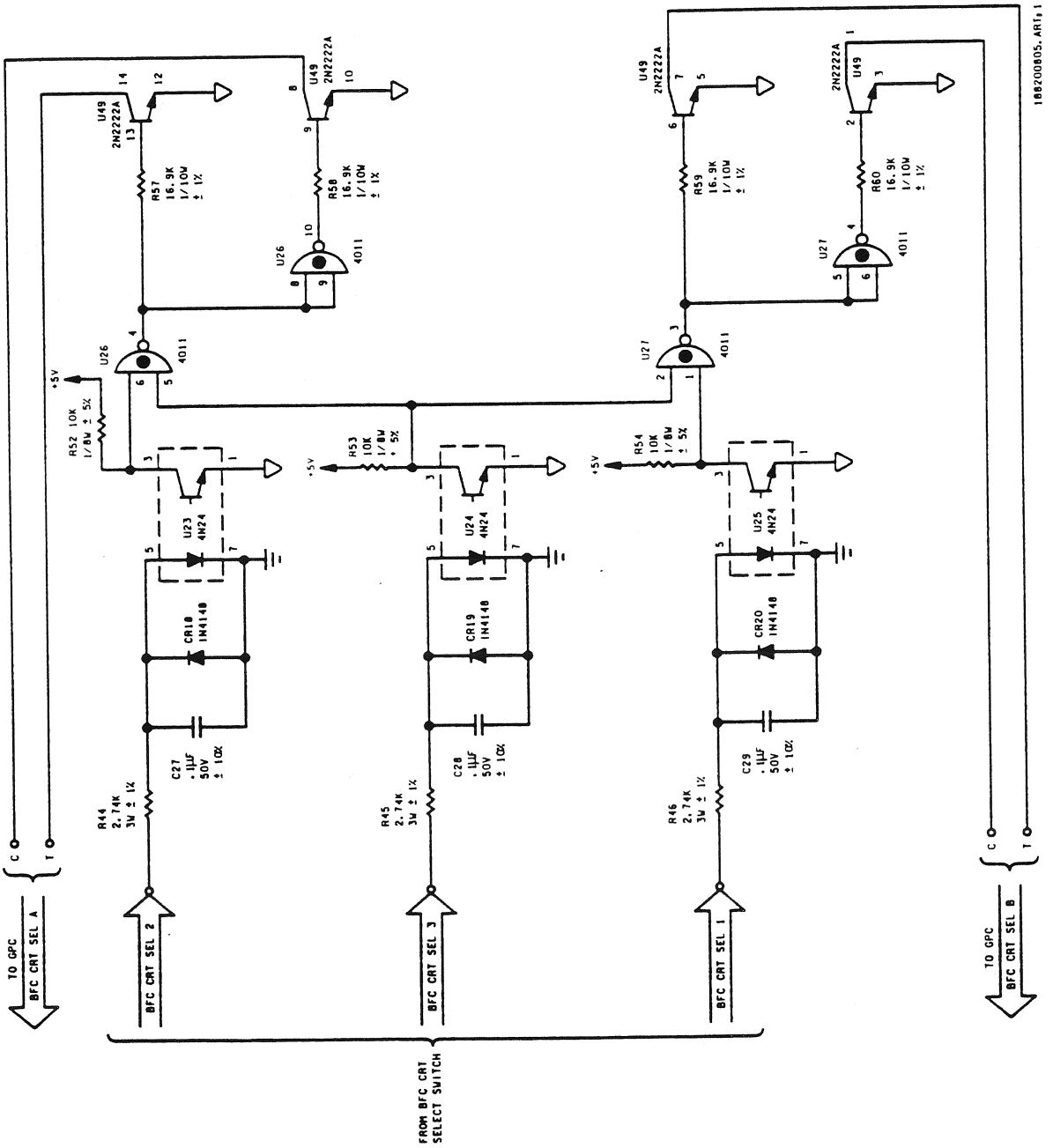


Figure 8-5.- BFC CRT select logic.

We also normally think of the T/C pair as providing a true differential interface; i.e., equal and opposite current flow in each side of the pair. Again this is not really the case. For our system, a positive logic signal is present when the "true" side is grounded or "closed" and the "complement" side is ungrounded or "open".

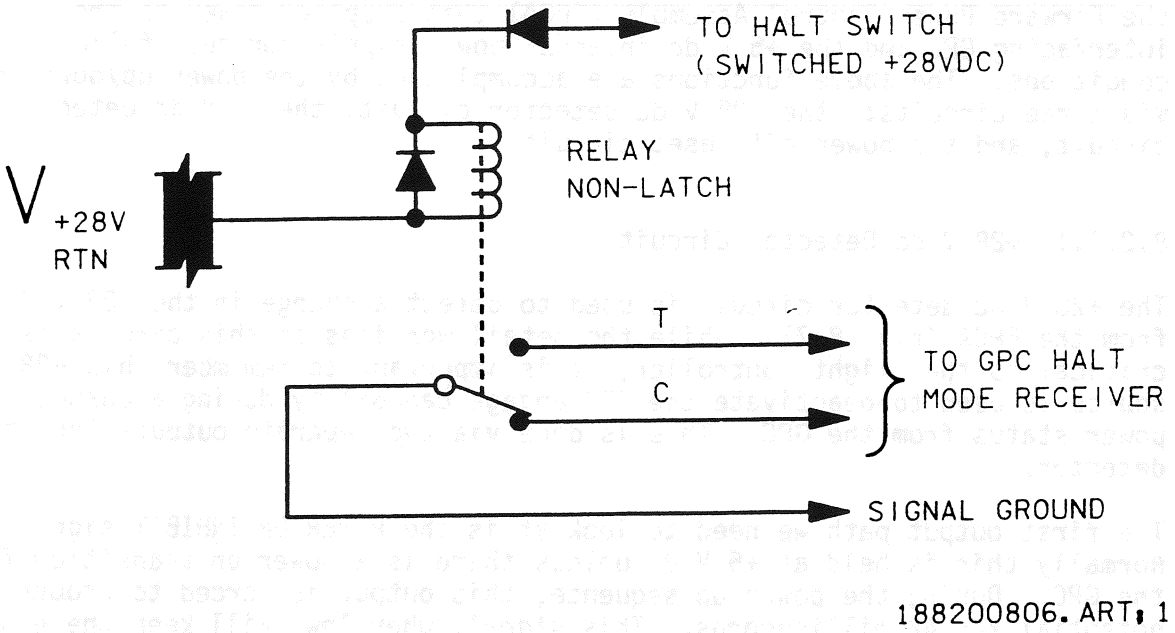


Figure 8-6.- Halt relay.

8.2.3 Power Up/Down Logic

The power up/down logic provides several functions for the BFC. First, it provides a stable output (inhibits the engage capability) from the BFC during the power transitions times of the BFC. Secondly, a signal is supplied to the engage/disengage logic to reset it to a disengage state if the power is turned off. Lastly, and really a part of the first two functions, the power up/down logic monitors the +28 V dc input power from the Forward Power Control Assembly (FPCA) that supplies power to the interfacing GPC and the +5 V dc internal power supply for out-of-limit conditions. The above functions are accomplished by the power up/down logic via three circuits: the +28 V dc detector circuit, the +5 V dc detector circuit, and the power off reset circuit.

8.2.3.1 +28 V dc Detector Circuit

The +28 V dc detector circuit is used to detect a change in the +28 V dc from the FPCA (fig. 8-7). While the detail workings of this circuit is not critical to the flight controller, it is important to remember this +28 V dc source is used to deactivate the BFC engage capability during a change in power status from the GPC. This is done via two separate outputs from the detector.

The first output path we need to look at is the POWER ON INHIBIT signal. Normally this is held at +5 V dc unless there is a power on transition from the GPC. During the power up sequence, this output is forced to ground potential for 42 milliseconds. This signal, when low, will keep the engage circuits from being active at the output of the BFC. The 42 milliseconds is needed to keep the BFC from sending out false engage signals while the +5 V dc logic power is stabilizing to a full on state.

The second path sends a signal related to the GPC power to the POWER OFF RESET circuit. The information is used to determine a power off transition or a fluctuation of the GPC power source. The signal goes high if the the +28 V dc drops to +22 V dc for more than 402 microseconds. If you look at the vendors drawing, you will find two signal lines coming from the +28 V dc detector to the power off reset circuit. The second line is a latch signal which is used to tell the power off reset circuit when the signal from this module is valid.

8.2.3.2 +5 V dc Detector Circuit

The +5 V dc detector circuit shown in figure 8-8 is used to monitor the +5 V dc internal power supply. The output from the comparitor circuit is sent to the Power off reset circuit. This signal is held low if the +5 V dc level is less than +4.5 V dc or greater than +6.4 V dc which allows for some normal fluctuations in the power supply output.

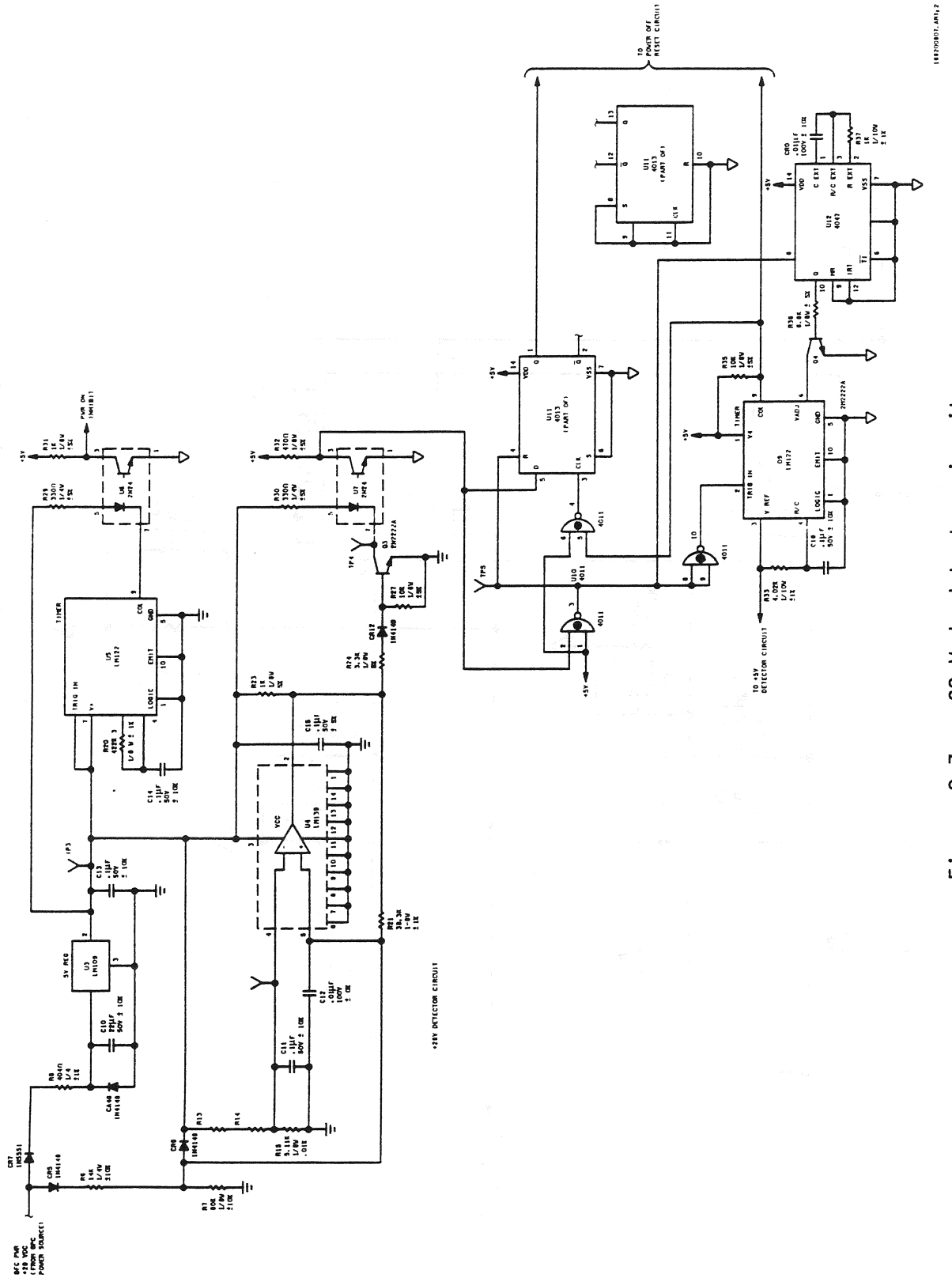


Figure 8-7.- +28 V dc detector circuit.

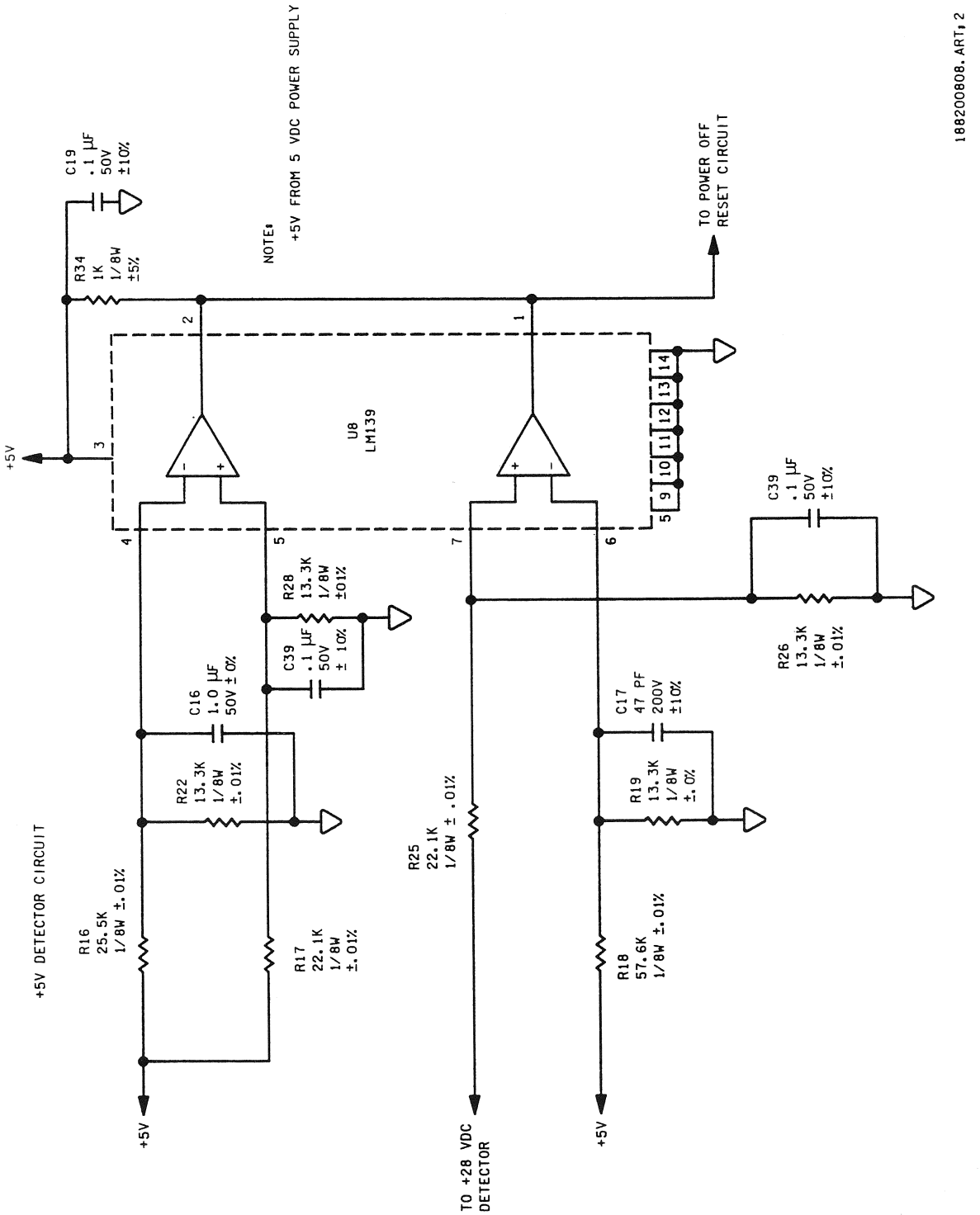


Figure 8-8.- +5 V dc detector circuit.

8.2.3.3 Power-Off Reset Circuit

The Power-Off Reset Circuit function shown in figure 8-9 is to reset the engage latches if the +28 V dc power has been on (PWR-ON INHIBIT is high) and power from the +5 V dc and the +28 V dc detector circuits are out of tolerance for more than 402 microseconds. The PWR-OFF RESET signal from this circuit is normally high. When the circuit gets the information from the detector circuits that an out of tolerance voltage condition exist, the PWR-OFF RESET goes low, resetting the engage logic latches to the non-engage state. The PWR-OFF RESET DELAY LATCHED signal is normally high and will drop and latch low if the out-of-tolerance power condition is detected for more than 2.5 milliseconds.

As a conclusion to this discussion, the Power Up/Down Logic in the BFC monitors the quality of the power provided from the GPC. If the state of the power is questionable, the BFC assumes the engage information inside the BFC is suspect; therefore, the engage capability is inhibited. The drawing in the Systems Handbook (8.2) summarizes the functional output of this circuit. For that drawing, the PWR OFF RESET is normally logic 0. The PWR ON INHIBIT and the PWR OFF RESET DELAY are normally logic 1.

8.2.4 GPC Select Logic and Delay Filter

The function of this part of the BFC circuitry is to control the state of the I/O TERM B output based on the information from the GPC OUTPUT switch and engage state as well as control the BFC engage capability. There are two signals from the GPC OUTPUT switch which are used to dictate the BFC reaction to an engage signal, the BFC GPC SELECT and BFC GPC SELECT OFF.

The BFC GPC SELECT signal line is isolated from the rest of the BFC via an optical isolator. The signal is then sent through a circuit which is shown on the functional handbook drawings as a delay filter. This circuit as shown in figure 8-10 is really a pulse trigger (LM 122) integrated circuit and a D flip-flop. The delay is used to validate the incoming select signal while holding the output steady. This is done by latching the signal value at the output (Q) of the D flip-flop. When a new state is received, the pulse trigger will "fire" a pulse to the D flip-flop circuit clock input such that the value at the input (D) is transferred to the output. Because of the network design, the pulse trigger is not sent unless the change in voltage at its input is high enough and long enough in duration to cause the trigger action.

From the filter network, the signal is combined with the engage signal through an exclusive OR gate, which then forms the I/O TERM B signal as shown in figure 8-10. The I/O TERM B signal is therefore dependent on the condition of the ENGAGE signal and the state of the SELECT signal for the interfacing GPC. If the GPC SELECT is a logic 1, the interfacing GPC is the BFC GPC or an SM GPC. During a nonengage period, the I/O TERM B will be set. If either signal is changed, the I/O TERM B will be dropped to allow the interfacing GPC to communicate on the FC Bus. (See the system brief on the GPC and the BFS operations for more information on the I/O TERM B.) The

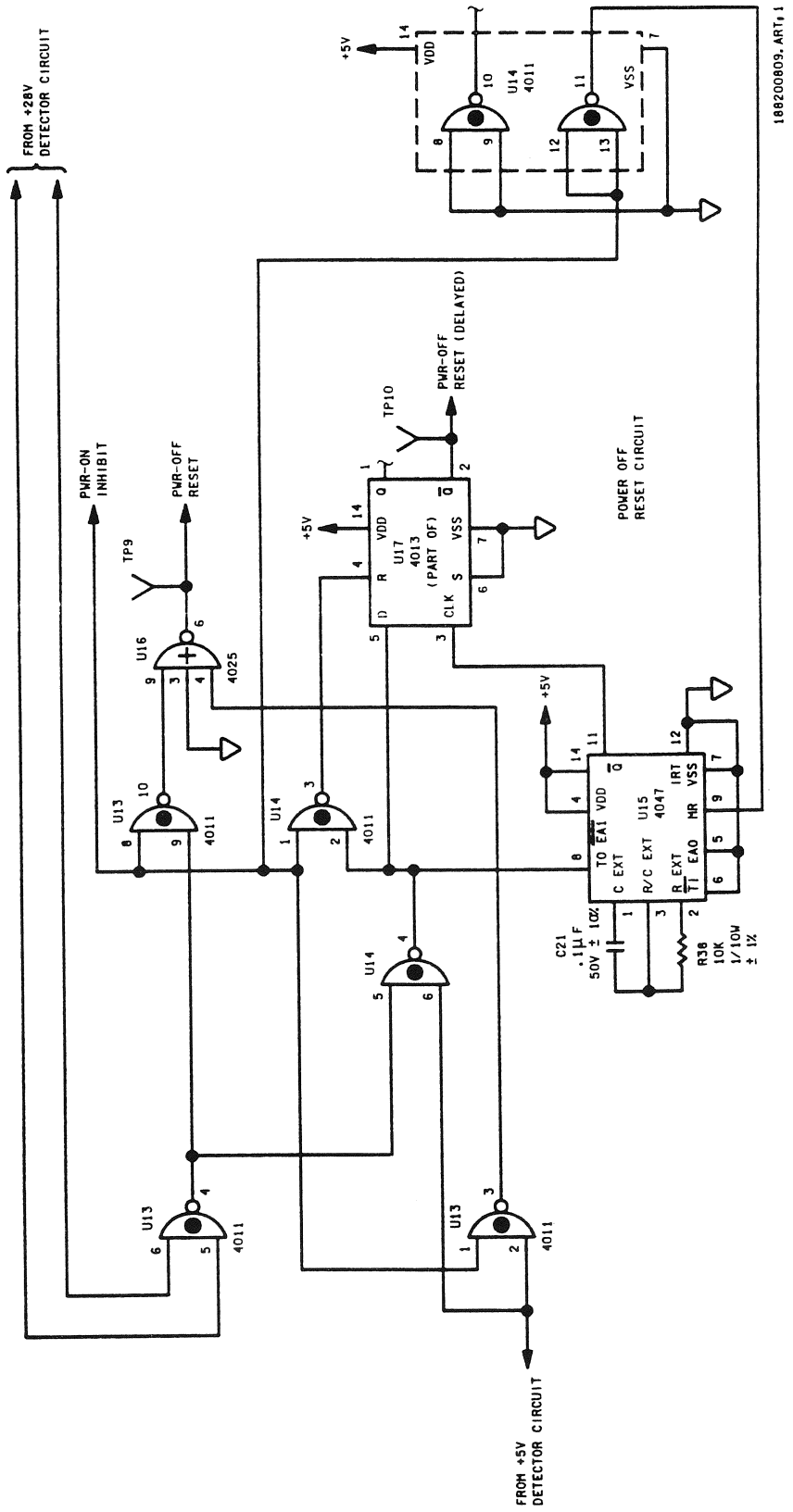


Figure 8-9.- Power-off reset circuit.

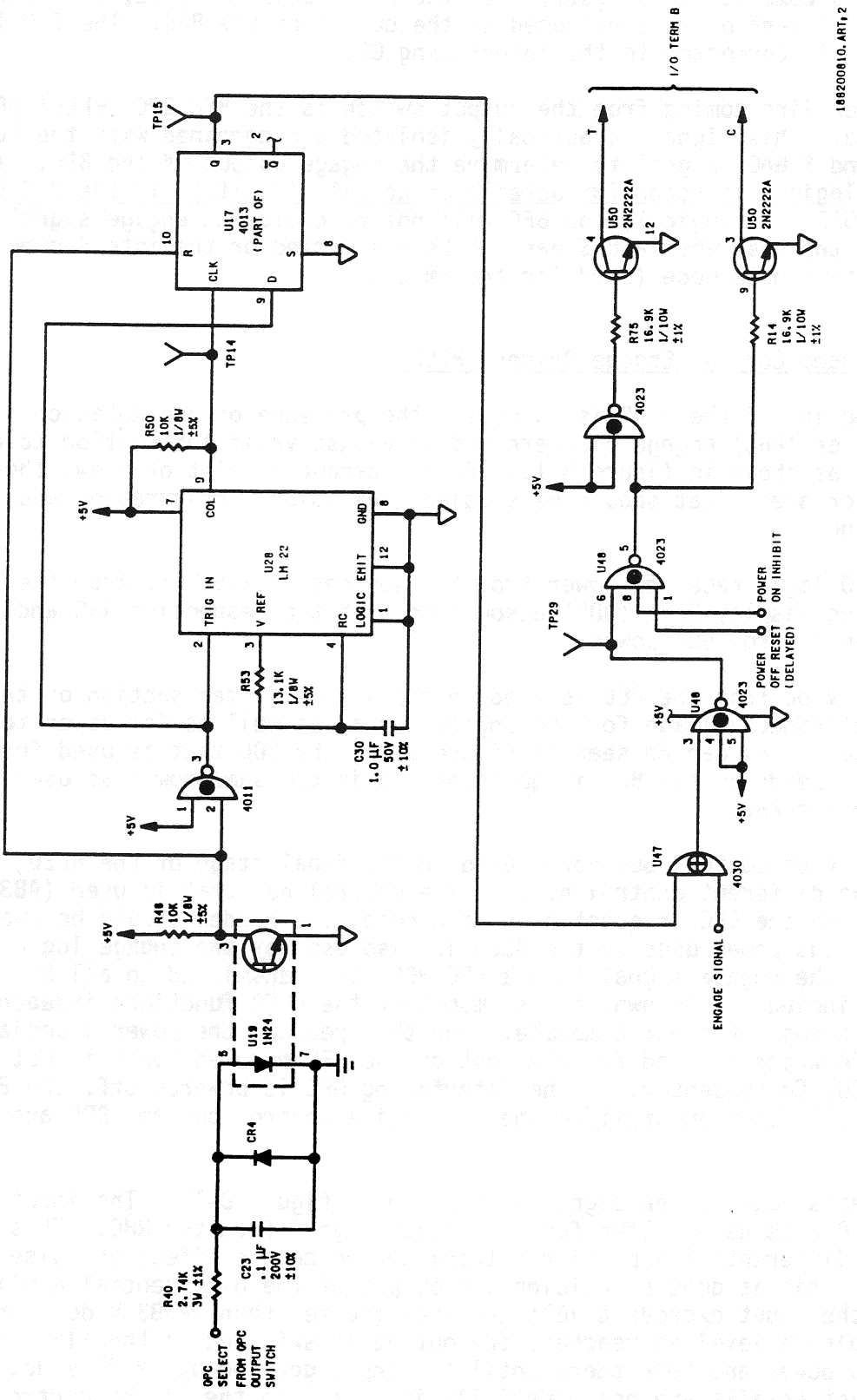


Figure 8-10.- I/O TERM B logic flow.

signal is combined (NAND gate) with the POWER UP/DOWN logic, and then a true/compliment pair is produced at the output of the BFC. The I/O TERM B, T/C pair is connected to the interfacing GPC.

The other line coming from the output switch is the BFC GPC SELECT OFF discrete. This signal is optically isolated and combined with the POWER OFF RESET and ENGAGE signal to determine the engage output of the BFC. (See the engage logic discussion for details of actual circuit.) If the BFC GPC SELECT OFF is a logic 1, the BFC will not recognize an engage signal. This will be the case when a BFS has not been selected or the interfacing GPC is in the terminate mode (used for the SM GPC).

8.2.5 Hand Control Engage Driver (HCED)

The function of the HCED is to detect the presence of one Rotational Hand Controller (RHC) engage discrete and broadcast valid information to all BFC modules as shown in figure 8-11. From a hardware point of view, there are two major areas that should be studied, the power configuration and signal filtering.

The HCED logic receives power from two sources, a +15 V dc from the Dedicated Display Unit (DDU) associated with the respective RHC and +28 V dc from the control bus power.

The +15 V dc from the DDU is used in the signal filter section of the HCED as a reference voltage for the engage signal as well as for power to the HCED logic. As can be seen in figure 8-11, the DDU that is used for the BFC module depends on the BFC in question and is the same power as used by the interfacing RHC.

The +28 V dc control bus power used in the final stage of the HCED, comes from two different control buses. The control bus that is used (AB3 or CA1) depends on the BFC in question. As a note, the reader should be aware the control bus power used by the HCED is also used by the engage logic. Because the engage signal from a BFC HCED is transmitted to all BFC engage logic, including its own, it is important the HCED functions independently from the rest of the BFC module. For this reason, the power associated with the FPCA which is used for the rest of the BFC modules logic is not used by the HCED. Consequently, if the interfacing GPC is powered off, the BFC HCED will still function provided the respective control bus and DDU are powered up.

Now, let's look at the signal path shown in figure 8-12. The input stage of the HCED acts as a filter for the engage signal from the RHC. This circuit uses a differential amplifier network to reduce the effect of noise at the input. This is done by holding the output of the differential amplifier low until the input exceeds a voltage level greater than +2.83 V dc. When the trip voltage level is reached, the output is switched to the +15 V dc from the DDU power and held there until the input goes below +2.37 V dc. While the exact details are not especially important to the flight controller, it

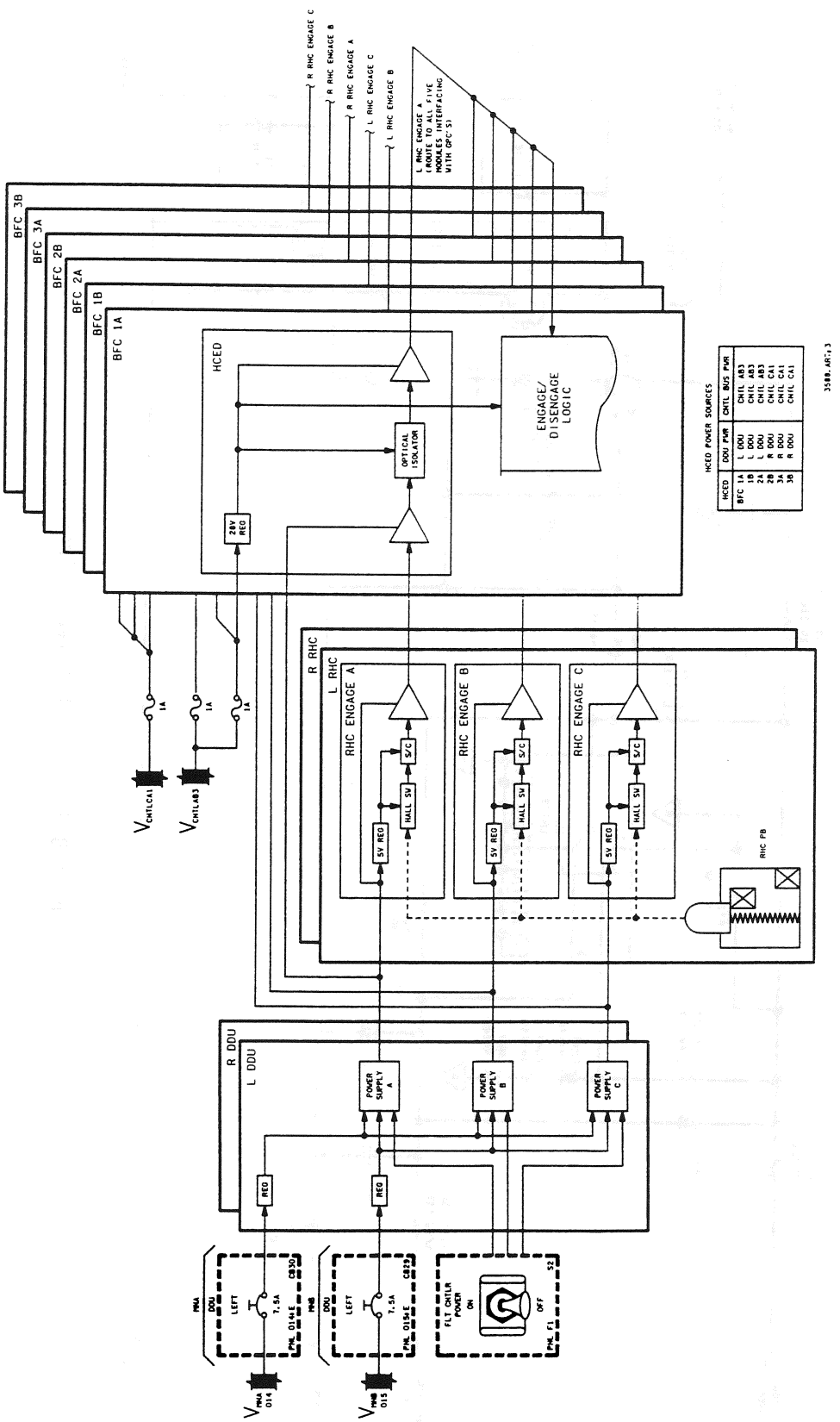


Figure 8-11.- HCED/DDU/RHC interface diagram.

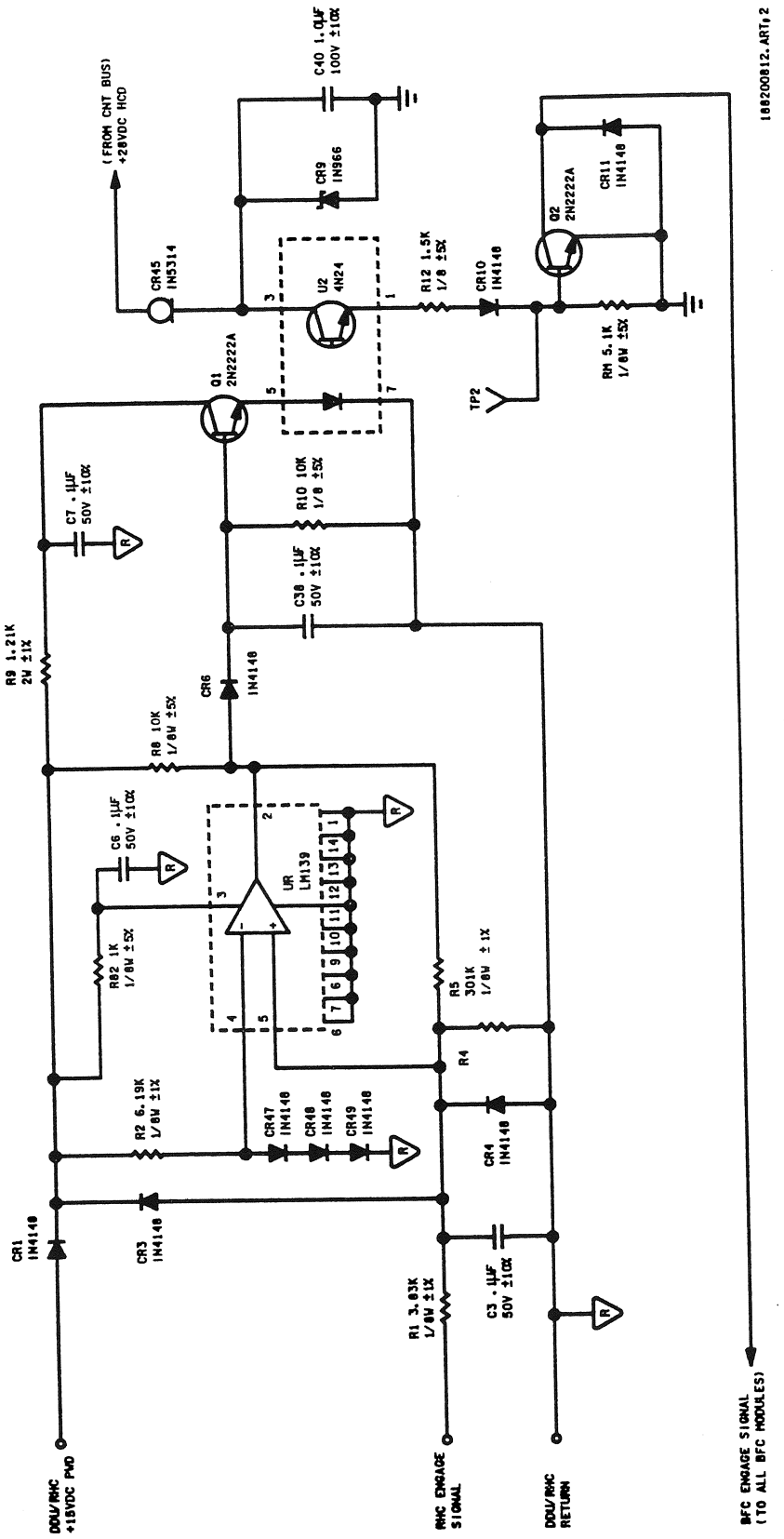


Figure 8-12.- HCED driver circuit.

is significant enough to note that noise on the engage lines is filtered out so that an inadvertent engage is avoided.

The filter signal is then used to drive an optical isolator. At the output of the optical isolator, the +28 V dc from the control bus is used to control the output switching transistor (Q2). The output of the HCED is in reality an input from the engage control logic, and functionally operates like the halt relay. During a non-engage state, this output switch is held open. When an engage signal is detected, the input of the optical isolator is activated, which allows the +28 V dc to activate the output switching transistor, which then applies a ground potential at the output of the HCED.

8.2.6 Engage/Disengage Logic

The function of the engage/disengage logic is to provide the proper state of the engage signal to the interfacing GPC based on the variety of information provided to the BFC. The decision to send a change in state of the signal to the GPC depends on many factors. As discussed earlier, the BFC provides several internal filters and stabilization networks to keep the engage signal as stable as possible. The output of those circuits are combined here with the ENGAGE discretes from the HCEDs and DISENGAGE discretes from the BFC DISENGAGE switch.

You should notice from prior discussions and the discussion to follow that most of the BFC design characteristics are used to prevent the interfacing GPC from receiving an improper "active" engage signal. This was done because the problems caused by an inadvertent engage would be worse than one GPC not engaging. This is true up until the crew needs to engage, and the BFS GPC BFC is the one that cannot engage. Given the probability of needing to engage and then having the BFC not respond is very low; it is a risk that is accepted in the current design.

There are six separate engage inputs to each BFC module and three disengage inputs. Both types of signals have an optical isolator at the input, but the configuration and power source are not alike (fig. 8-13).

The ENGAGE optical isolator utilizes the +28 V dc from the control bus (also used by the HCED) to drive the input side of the isolator. If the BFC is engaged, the discrete input is grounded (at the HCED) causing the isolator to activate the output. By looking at the output of the isolator, we can see that it is held at +5 V dc until activated at which time the line is pulled low (grounded).

The DISENGAGE signal optical isolator is configured differently. The input to the DISENGAGE isolator requires +28 V dc to be supplied as the signal instead of the ground provided for the ENGAGE isolator. The output of this isolator is the same as the ENGAGE isolator except the DISENGAGE signal is split into two signals to now match the ENGAGE set of signals (six).

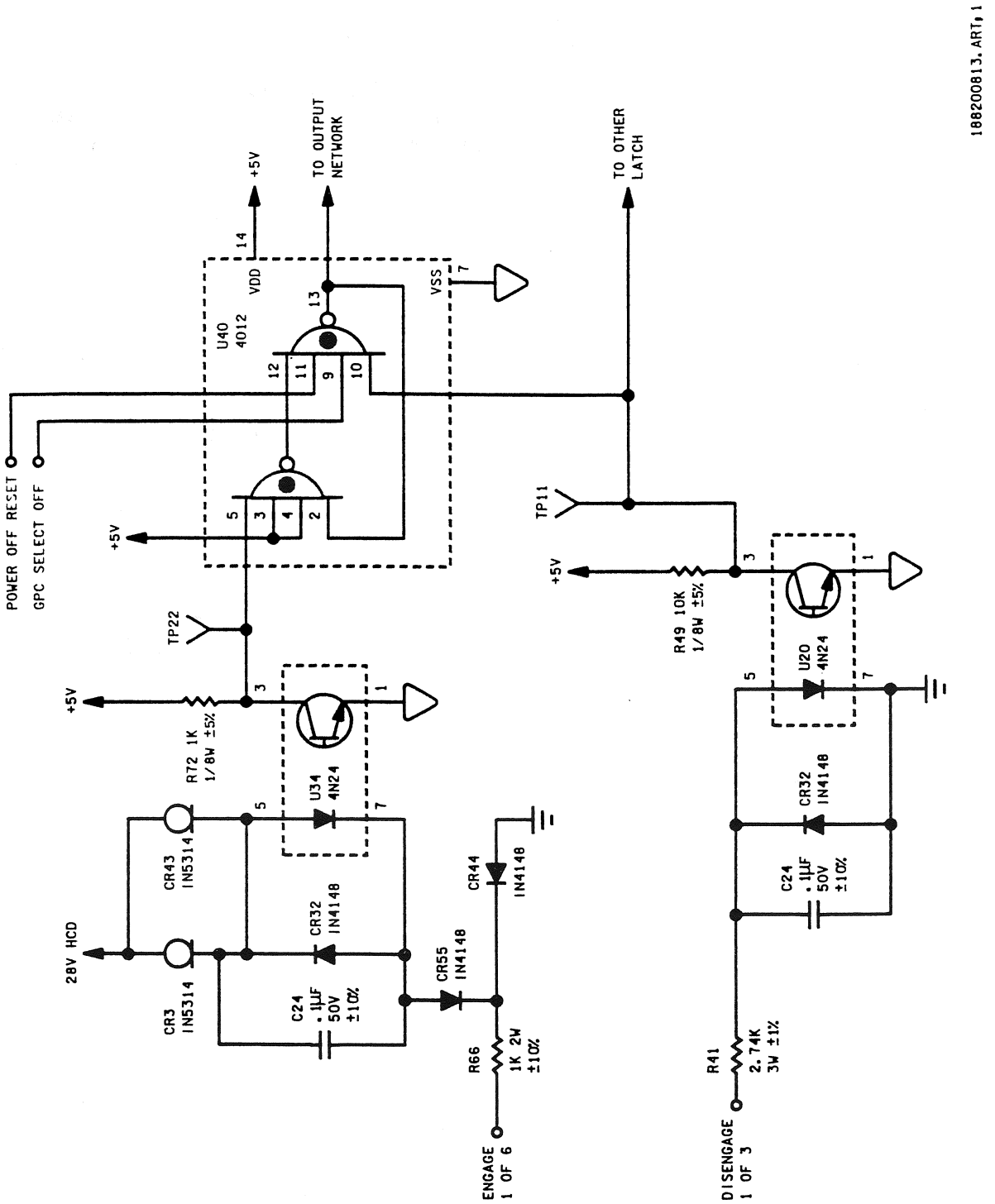


Figure 8-13.- ENGAGE/DISENGAGE signal inputs.

The ENGAGE and DISENGAGE logic signals are combined at a latching network which is configured similarly to a SET/RESET type of flip-flop. The network is an IC consisting of two NAND gates with four inputs each. Information from the GPC SELECT OFF and the POWER OFF RESET are also combined with the engage logic at this network.

The input of the network is the ENGAGE signal. This logic signal is NANDed together with the output of the network (ignore the +5 V dc input to this gate, it is only used to keep those inputs high to disable their effect). The output of the first gate is combined with the DISENGAGE, POWER OFF RESET, and the GPC SELECT OFF signals. Notice that each of the three "other" signals act as network reset type inputs. This is true because they are normally held high so the engage signal controls the action.

As stated earlier, the output of the network wraps around and is tied to the input NAND gate. This is the "latching" signal. When this network is in a disengage state, the input and output is in a high state. When an "active" engage signal is received, the input is set low, causing the first gate output to become high, causing the output of the second gate to become low, which then wraps around to apply a low input for the first gate before the momentary "active" engage signal returns back to the "inactive" high state.

Sense it requires only one low input to keep it in the engaged state, the network will latch (much like a D Flip-Flop) to the engaged state until the output NAND gate is forced to change states due to a change of one of it's inputs. Typically, the signal to unlatch the network will be the DISENGAGE signal (unless there is a power problem).

From the latching network, the signal is then sent to the output network shown in figure 8-14. This network is a series of logic gates that combine the six individual ENGAGE signals, makes four logic signals (three engage and one I/O TERM B), makes a final check of the power condition, and then split each signal into a true compliment pair for transmission to the interfacing GPC.

As stated earlier, there are six separate engage signals being processed by the BFC, three from the right RHC and three from the left RHC. The set of signals from each RHC are checked against each other for validity; there must be three ENGAGE signals from the RHC in question (at this point in the circuit, a low logic signal is an "active" engage signal). This check is in the form of a NOR gate which produces a high output if and only if the three inputs are low.

The signal is then combined with the result of the other RHC to form the engage information to the I/O TERM B logic output. If either RHC produced the necessary three ENGAGE signals, the output of this gate will be low (active engage). Then, depending on the state of the GPC select signal, the proper I/O TERM B signal will be sent (as was discussed in section 8.2.4).

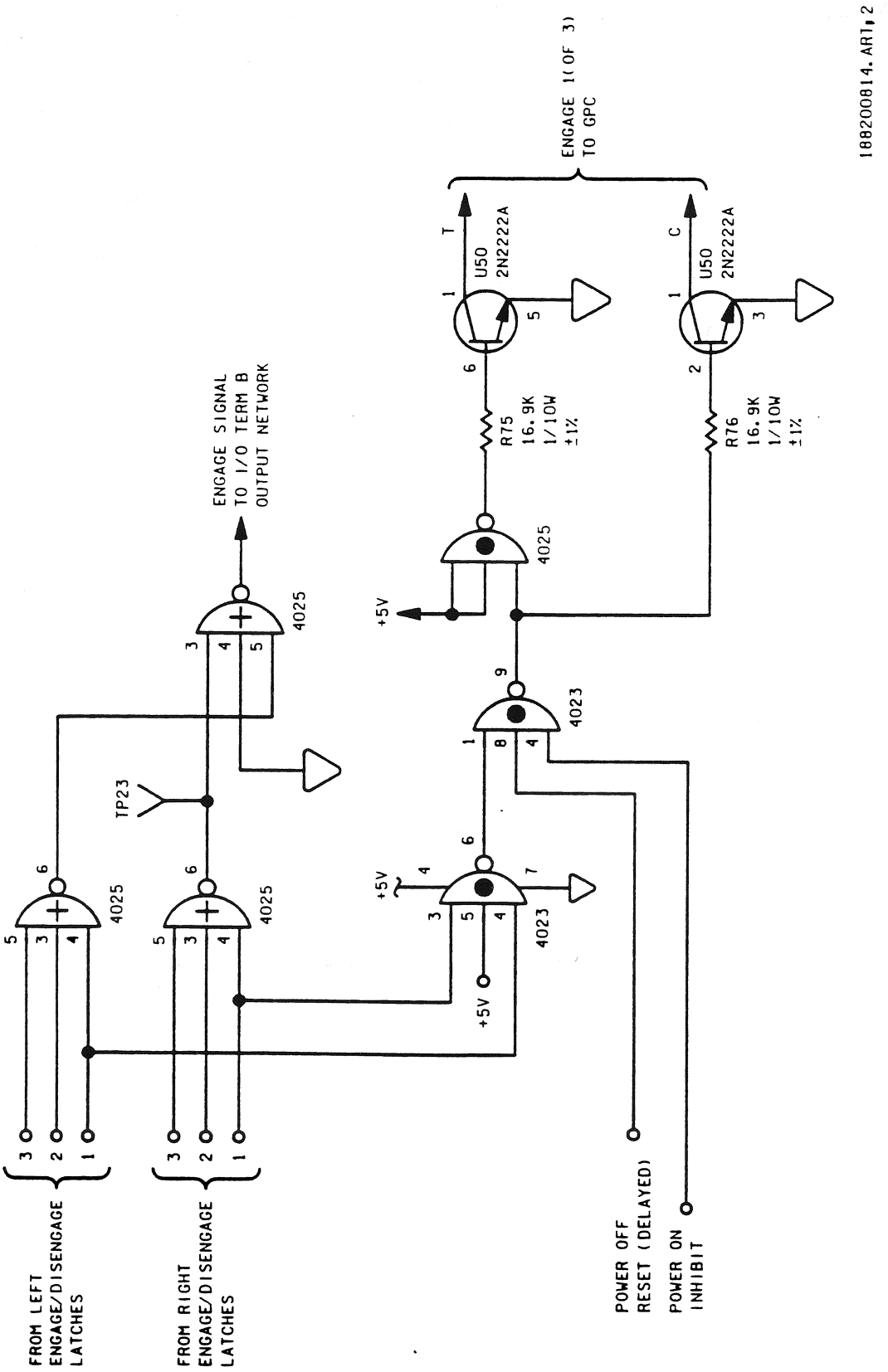


Figure 8-14.- ENGAGE signal output.

The three engage output signals are produced without checking the redundancy from each separate RHC discrete as was done for the I/O TERM B. A signal engage discrete from each RHC is combined to form one of the three engage outputs signals. If either input to the NAND gate is an active engage signal (low), then the output from this gate will go to the high state.

This signal is then combined with the power indicators as mentioned earlier, split into a T/C pair, and sent to the interfacing GPC.

If you examine the BFC functional drawing in the Systems Handbook (8.2), all of the signal lines are converted such that the logic signal is high for an active signal. It is a good idea to review the handbook drawing at this time as a good review of the BFC as a complete box.

Any one of the five GPCs (or none) can be selected for the BFS GPC. As mentioned previously, the BFS GPC selection is accomplished by placing the OUTPUT switch in BACKUP. The BFS hardware inhibits response for approximately 1 second when the switch is moved. See figure 8-15 for a diagram of the PASS/BFS GPC discrete interfaces.

8.3 PASS/BFS INTERFACE OVERVIEW

BFS is functionally independent of PASS after engagement. BFS is activated by depressing the downmode button on either RHC (see section on discrete interpretation). BFS is a multi-sensor, single-string system for GNC functions and sequencing; BFS also performs nonredundant Systems Management (SM) and Special Processes. During normal Ascent/Entry phases, BFS is the only source of SM, and the BFS SM is somewhat limited in relation to the PASS SM.

Prior to BFS engagement, the required sensor and crew input data to perform several functions is obtained by "listening" to PASS commanded data on the Flight Critical Display and Display Keyboard (DK) data buses. The PASS also transfers data to the BFS (defined in ICD) for the BFS initialization and tracking.

The BFS's perception of the state of the vehicle is determined with the same sensors and crew inputs as PASS; however, BFS has different and independent operating and application programs from PASS.

BFS functional requirements are the same as PASS with some key exceptions:[2]

A. Different operating system to preclude generic problems with PASS

B. Multi-sensor with simple Redundancy Management (RM) single-string system

C. Limited on-orbit capabilities (vehicle attitude control, SM/SP and P/L, FDA)

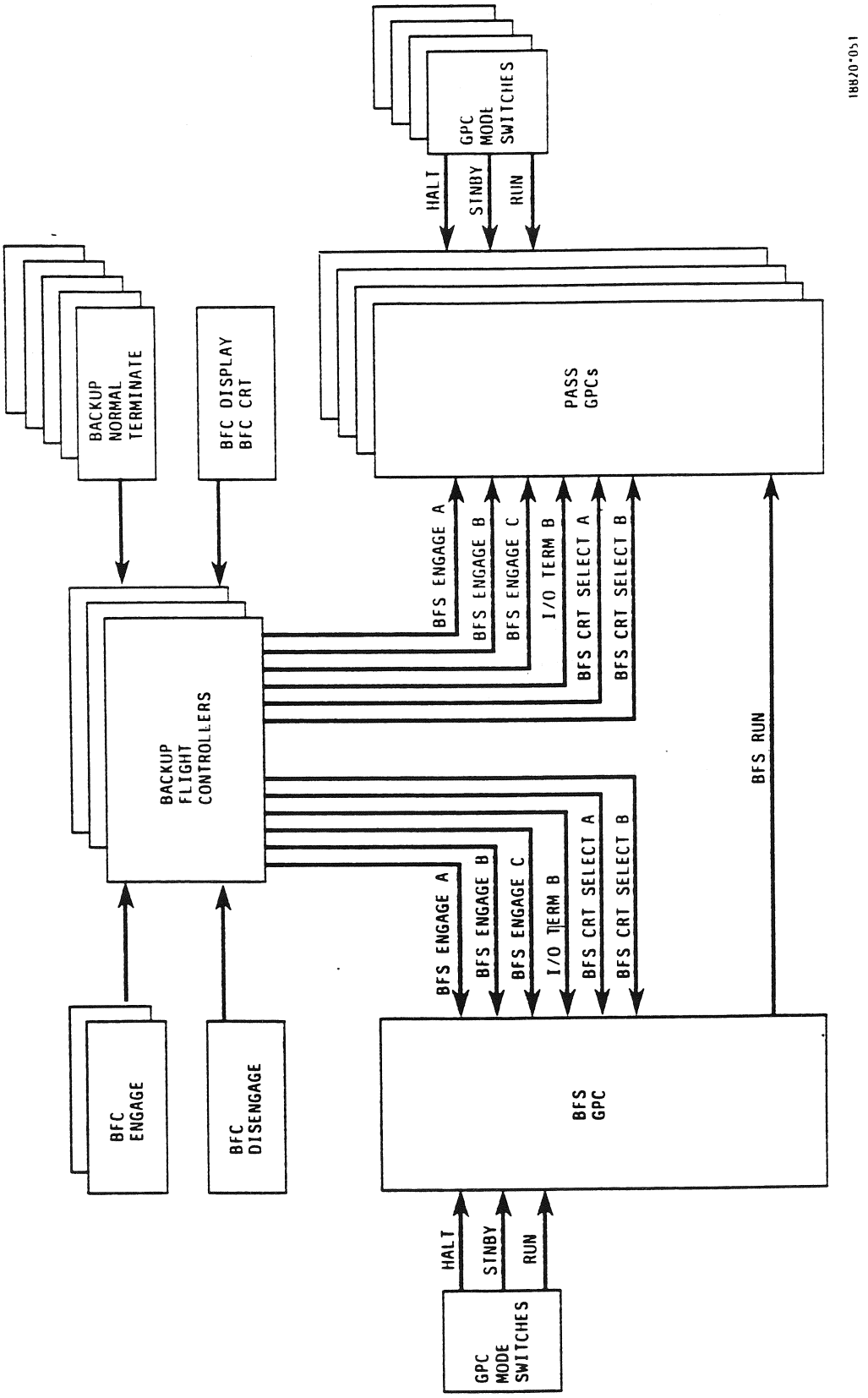


Figure 8-15.- PASS/BFS GPC discrete interface block diagram.3

D. Reduced GNC capabilities:

1. No preflight or on-orbit IMU calibration and alignment
2. No MSBLS data
3. Radar altimeter processed for display only
4. Ascent guidance until MECO is auto only, all other guidance is with a man in the loop
5. Guidance command computation during entry is terminated below 2000 feet altitude with respect to the runway
6. Control Stick Steering (CSS) landing similar to OV-101 BFCs
7. Data Processing for Heads Up Display (HUD) Electronics not supported

E. Reduced SM/SP functional requirements

F. Launch Bus interface which includes protocol for GPC read/write only

Please see the BFS Software Function diagram (fig. 8-16) for a graphic illustration.

No dynamic switchback to PASS during Ascent/Entry can occur if BFS is engaged; however, BFS can accommodate a switchback during ground checkout phases, as well as during on-orbit phase.

8.3.1 Disengaged/Engaged Manual Ops Transition/Initialization

In order to transition from OPS 0 to 101 or 301, BFS requires external inputs from subsystems (RGA, AA, IMU) and PASS transfer data. If BFS is disengaged and tracking PASS, then transition from OPS 0 to 101 or 301 is legal. When the transition is made, PASS should be operationally ready in 101, 301. Therefore, the transfer data must be provided.

If BFS cannot establish PASS tracking and BFS is disengaged, the BFS should be inhibited and appropriate messages will be annunciated (see JSC Document STS 83-0020 - The D&C FSSR).

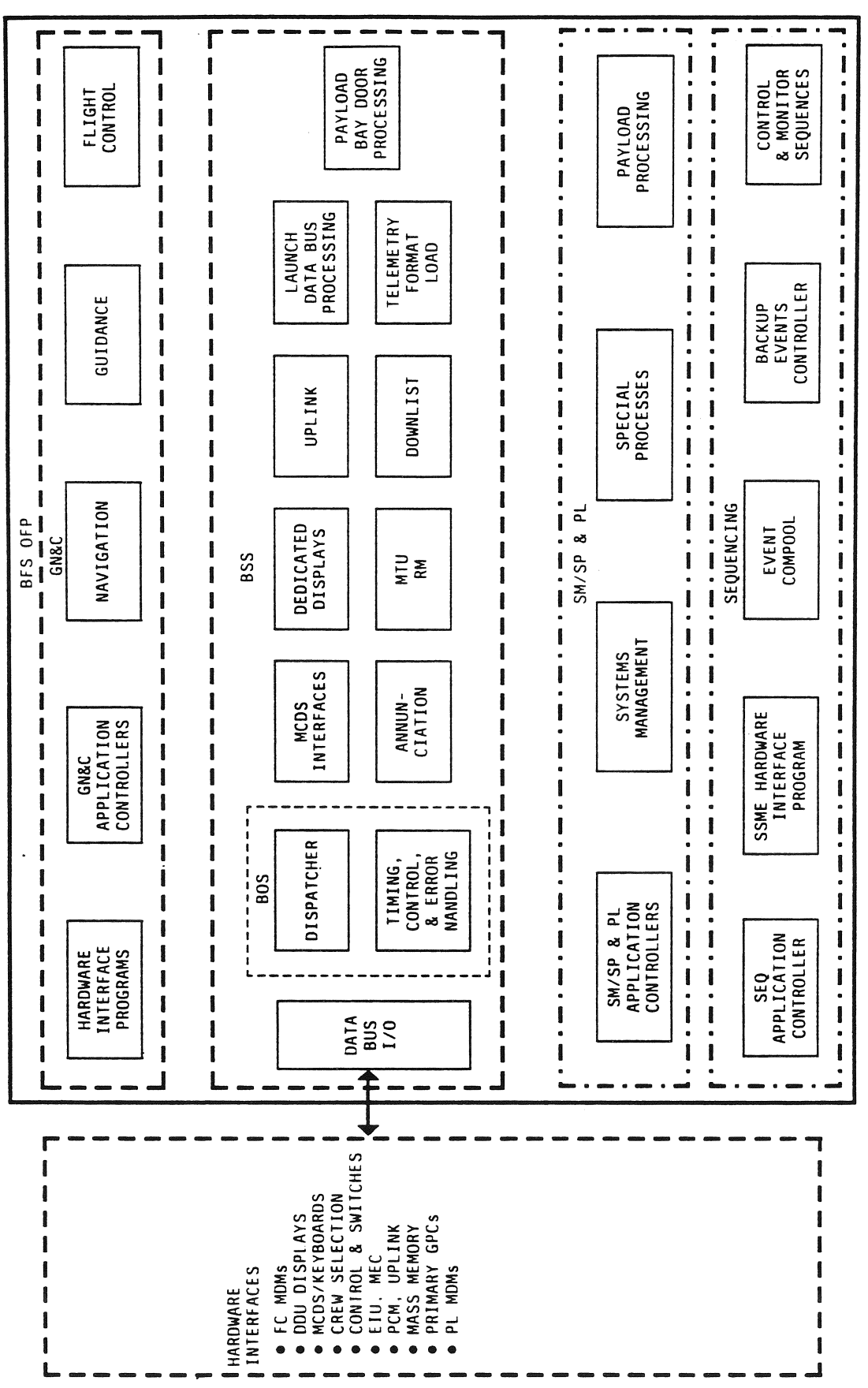


Figure 8-16.- BFS software function.[2]

Tracking PASS is based on the status of four flight critical strings; BFS makes no association between flight critical strings and PASS GPCs. During OPS 1, 3, and 6, PASS transmits 25 Hz transfer data once per minor cycle on flight critical buses 5-8 (this corresponds to strings 1-4). The data from all enabled strings is compared following the input, and BFS continues to listen on those strings whose transfer data agrees. If two independent sets of agreeing strings are determined through comparison of the transfer data (i.e., for a 2 on 2 redundant set split), BFS sustains tracking on string 1 and its set partner. Therefore, from a BFS tracking standpoint, an operational preference would be to assign strings 3 and 4 together, or strings 2 and 4 to the same GPC if it becomes necessary to establish a 2 or 3 primary GPC redundant set.

If BFS is engaged, both transitions are legal; however, the BFS shall inhibit outputting on the Flight Critical buses during 101 engaged. The transition to 301 is permitted while engaged in order to provide BFS attitude control of the vehicle assuming appropriate LRUs are operational.

8.3.1.1 OPS 0 Processing[2]

The following information relates what BFS processes are operational while in OPS 0:

BSS
- Data Bus Configuration
- Disengage PASS tracking initialization
- Uplink (if NSPs inputs available)
- Downlink (ascent format)
- One shot transfer via keyboard entry on the GPC Memory display

D&C
- GPC Memory and Fault Displays

GN&C
- Applications Controller

SM/SP/PL & Sequencing
- BEC
- SM/SP/PL & Sequencing Applications Controllers
- SM vice antenna management
- Special Processes
- RCS quantity gauging

The following processes are unique to BFS SM as opposed to PASS SM:

- Cabin Pressure monitoring
- Freon Pump management
- FES/NH_3 boiler act-deact

8.4 BFS AND PASS MCDS INTERACTIVE PROCESSING

The BFC plays an important role in controlling the operation of the PASS and BFS computer. As discussed in the hardware section, many safeguards have been built into the design to prevent single failures from inadvertently engaging the BFS and/or halting one or more PASS computers. On the other hand, the BFS by design is a singular system and therefore contains many single point failure locations which can prevent the BFS engage operation. In addition, the BFC modules receive power from three sources; any one being down impacts the operations of one or more modules. The operations and failure mode impacts are discussed in the following subsections.

8.4.1 General Characteristics

A. BFS provides CRT display information on a single forward display selected by the crew while in preengage mode. On engagement, BFS provides display information on two selected forward CRTs. Preengage, PASS provides display information on the DDUs and Special Processes because PASS has control of the Flight Critical buses. On engagement, the BFS drives the DDUs and the SPI. The BFS also drives the aft station CRT.

B. BFS and PASS sharing of the CRTs is a function of the BFS CRT SELECT switch, the BFC CRT select logic, BFS preengaged/engaged state, IPL SOURCE MMU switch, and GPC/CRT keyboard input. At this point, this section will discuss only the BFC CRT Display switch. This switch manipulates the states of the CRT SELECT A and B discretes from the BFC as highlighted in table 8-IV. Notice that if the BFC DISPLAY switch is off, the position of the CRT SELECT does not matter (during ENGAGE, the BFS will default to CRTs 1 and 2).

TABLE 8-IV.- BFS CRT SELECTION HIGHLIGHTS

BFC DISPLAY SW	CRT SELECT SW	BFS CRT(s) selected	
		Preengage	Engaged
ON	1 + 2	1*	1* and 2
ON	2 + 3	2*	2* and 3
ON	3 + 1	3*	3* and 1
OFF	N/A	None	1 and 2

*CRT bus is relinquished for GPC IPL if the IPL source select switch is in MMU 1 or MMU 2.

C. If a new Nominal Bus Assignment Table (NBAT) is invoked that reassigns a CRT which is currently assigned to the BFS via the BFC CRT Display switch, the new assignment will not be made until the BFC CRT Display switch is turned off. At that time, the previous PASS commander (from the old NBAT) will momentarily take that CRT until the new NBAT is invoked. If the BFC CRT Display switch is left on, the PASS cannot see the BFS CRT MF, resulting in no new NBAT assignment.

D. The possibility exists for BFS DK listen toggle-type item entries to get out of sync with PASS due to any number of causes, including entry on a BFS keyboard, re-IPL of BFS, differing requirements, I/O errors, and others. Subsequent entries of the item on a PASS CRT will toggle both systems, and will not make both systems consistent. To prevent opposite states of PASS and BFS DK listen toggle-type items, entries for these items should be made to a DEU commanded by PASS. If opposite states exist for a DK listen toggle-type item between PASS and BFS, the condition should be corrected by entering the correct state on the BFS display.

8.4.2 GPCIPL DEU Characteristics

A. Turning the BFC CRT display switch off is not sufficient enough to cause control to the DEU to be relinquished. To properly relinquish control of a GPCIPL DEU, one of the following must be performed:

1. Enter GPC/CRT XY EXEC where X is the GPC and Y is the DEU. X has to be 0 in order to free up the DEU.

2. Power down the GPC, then turn off the BFC CRT display switch to OFF.

3. Clear the message line prior to turning the BFC CRT display switch to OFF in order to cause control of the DEU to be relinquished, and control of the GPC to be given to SSL.

B. An error message may occur if keyboard entries are made on a DEU which is assigned to GPCIPL, but was last used by PASS and has not been re-IPL'd. If this occurs, the big X will stay on the screen, and the DEU will need to be IPL'd by toggling the DEU load switch.

Following the IPL, perform a message reset in order to clear the previous error/advisory messages.

8.4.3 Keyboard Processing

PASS and BFS keyboard item handling cannot be guaranteed to be identical. BFS DK listen response to PASS keyboard entries can be determined only by monitoring BFS display response. BFS may accept keyboard entries illegal in PASS. BFS may reject keyboard entries legal in PASS. (Note: BFS does not annunciate "ILLEGAL ENTRY" when it rejects an item entered via DK listen.)

8.4.4 DEU BITE

BFS will annunciate a "CRT BITE" message upon first occurrence of a DEU critical BITE indication in a DEU poll response. BFS will not reannunciate a CRT BITE unless it receives at least one good poll response without a critical BITE indication. BFS continues polling after a critical BITE and does not inhibit annunciation except for the expected critical BITE during DEU IPL. BFS operates differently from PASS in at least two respects: the critical BITE does not inhibit polling by the BFS, and the CRT BITE message is annunciated for DEU power transients.

8.4.5 SPEC Handling Nuances

In regard to the GPC Memory spec, BFS and PASS requirements differ on response to selection or reselection of engineering units or hex. PASS blanks data fields of the read/write portion of the GPC memory spec upon selection or reselection of engineering units or hex. Changing selection of engineering units or hex on the BFS Memory spec will cause the displayed value to update to the new units or format without the need for reentering addresses.

8.5 GPC HALT OPERATIONS

The HALT position of the GPC MODE switch is used to mode the GPC to a hardware controlled state where the electronics, discrete input and output drivers, and data buses are initialized to reset conditions.

Without the relay, possible electromagnetic interference on the halt wire could trigger the GPC discrete input driver and inadvertently halt the GPC. The location of the halt relay in the BFC was selected as a matter of convenience and is otherwise totally independent of the other BFC functions. If the BFC loses power, the halt relay will not be affected.

8.6 GPC SELECT OPERATIONS (GPC OUTPUT SWITCH)

The GPC OUTPUT switch controls the engage and PASS operation configuration by providing two discretes to the BFC module engage/disengage logic. There is one switch for each GPC, and all five switches are wired together as shown in figure 8-17. Depending on the switch position, the SELECT and SELECT OFF discretes will be either 0 V dc (logic 0) or +28 V dc (logic 1). Switch power is provided from Essential Bus 3AB and from the FCPA associated with the interfacing GPC (figs. 8-17 and 8-18). Note that a 3-amp fuse is between the ESS 3AB bus and the switches. The fuse is located on the back of panel O6 and is a possible candidate for In-flight Maintenance (IFM).

Any of the five GPCs can be selected as the BFS GPC by placing the appropriate GPC OUTPUT switch to BACKUP. If more than one GPC OUTPUT switch is in the BACKUP position, the switches are wired together such that the highest numbered GPC selected will be the BFS GPC. No BFS GPC is selected if all GPC OUTPUT switches are positioned to NORMAL or TERMINATE; i.e., the BFS engage function cannot be performed. Of course the successful engage action depends on the GPC that was selected as the BFS having the proper software loaded.

In the NORMAL position, the interfacing GPC will be able (provided the software is running) to perform normal GNC PASS functions because the GPC will be able to communicate on the flight-critical buses.

If the interfacing GPC will be utilized as the SM GPC, the output switch will be put into the TERMINATE mode and therefore will not be able to communicate on the flight-critical bus. It is important to note that if the BFS is engaged while a GPC is in the TERMINATE mode, the terminated GPC will not hear the ENGAGE command. To resolve the problem the crew will need to take the computer to NORM and reengage, or simply mode it to halt.

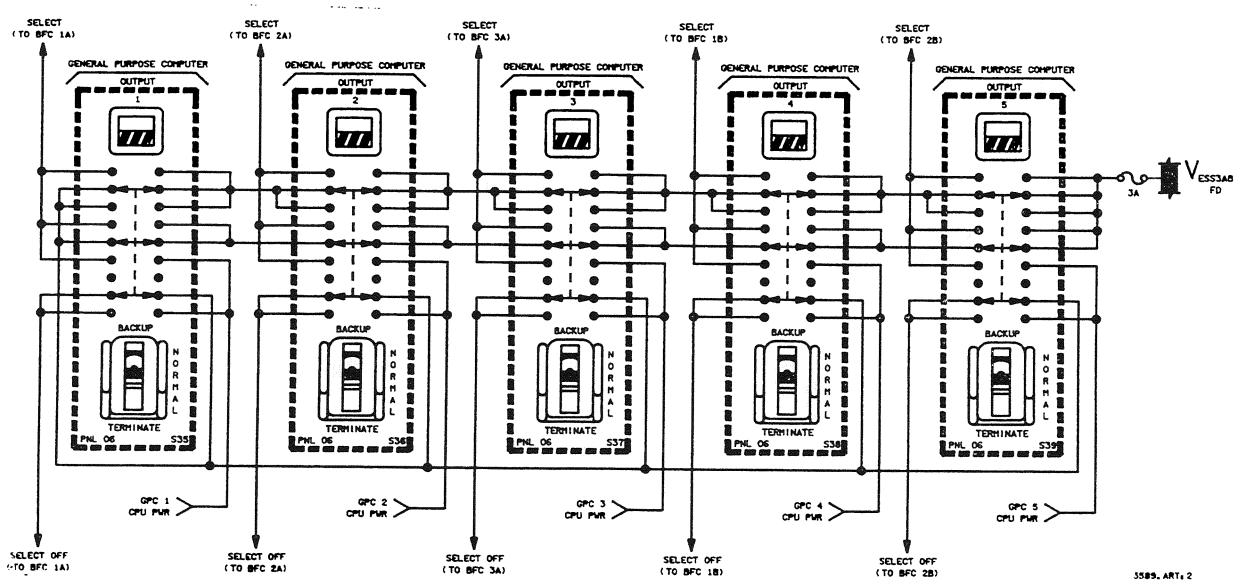


Figure 8-17.- GPC OUTPUT switch.

TABLE 8-V.- GPC OUTPUT SWITCH NORMAL CONFIGURATION

Nonimal ascent/entry configuration

OUTPUT sw / OUTPUT discretes	GPC 1 OUTPUT sw NORMAL	GPC 2 OUTPUT sw NORMAL	GPC 3 OUTPUT sw NORMAL	GPC 4 OUTPUT sw NORMAL	GPC 5 OUTPUT sw BACKUP
SELECT	0	0	0	0	1
SELECT OFF	0	0	0	0	0

NOMINAL ON-ORBIT CONFIGURATION

OUTPUT sw / OUTPUT discretes	GPC 1 OUTPUT sw NORMAL	GPC 2 OUTPUT sw NORMAL	GPC 3 OUTPUT sw NORMAL	GPC 4 OUTPUT sw TERMINATE	GPC 5 OUTPUT sw NORMAL
SELECT	0	0	0	1	0
SELECT OFF	1	1	1	1	1

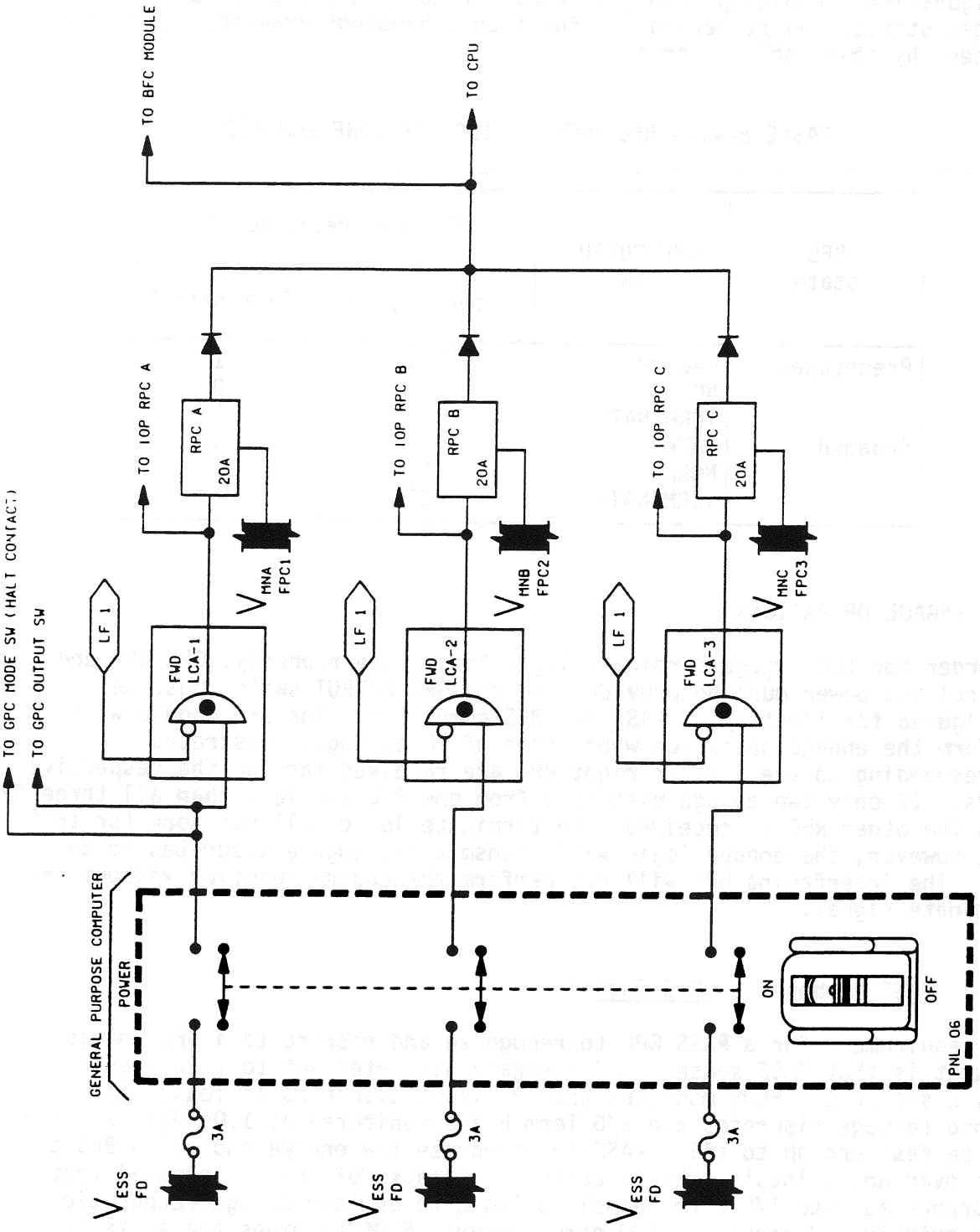


Figure 8-18.- GPC power switch interfaces.

Table 8-VI highlights the various engage and terminate B discrete configurations resulting from GPC OUTPUT switch positions and preengaged/ engaged states. By reviewing the functional handbook drawing, it can easily be seen why this table is true.

TABLE 8-VI.- BFC OUTPUT DISCRETE CONFIGURATION

BFC state	GPC OUTPUT sw	BFC discretes to GPC	
		Engage 1,2,3	Terminate B
Preengaged	BACKUP	000	1
	NORMAL	000	0
	TERMINATE	000	1
Engaged	BACKUP	111	0
	NORMAL	111	1
	TERMINATE	000	1

8.7 ENGAGE OPERATIONS

In order for the engage/terminate logic to operate properly, GPC CPU and control bus power must be provided and the GPC OUTPUT switch must be configured for identifying PASS and BFS computers. The BFC module will perform the engage operation when three of three engage discretes corresponding to the left or right RHC are received through the respective HCEDs. If only two engage discretes from one RHC and less than all three from the other RHC is received, the terminate logic will not work for that GPC; however, the engage logic will transmit the engage discretes to the GPC. The interfacing GPC will not perform the engage function without the terminate signal.

8.7.1 PASS Response to BFS Engage

The requirement for a PASS GPC to recognize and respond to a BFS engage request is that PASS senses 3 of 3 engage discretes set to 1, and an I/O Term B set to 1. FCOS monitors the GPC input discretes at least once per second (engage discretes and I/O Term B are monitored at 1.04 Hz), which may not be fast enough to allow PASS to recognize the engage and allow BFS to take over and maintain vehicle control. Because of this, a special test is incorporated into I/O error handling logic to perform engage recognition more quickly. Whenever an I/O error occurs, FCOS examines the state of the engage discretes and the I/O Term B (for an engage case, the I/O Term B will cause the I/O errors). If the engage criteria are met, the software will:

A. Issue an IOP rest

B. Light the CAM diagonal (I-fail)

C. Drive the output talkback to barberpole

D. Enter the WAIT state

Which results in the "classic" engage case requiring an IPL for recovery.

Several methods exist for engaging BFS without having the PASS go to software halt (one is presented here for its technical content, and the others are presented in the HIP POCKET IFM PROCEDURES section).

KSC tests the engage discretes by forming a 5 GPC OPS 0 common set with MDM FF1, FF2, and FF3 powered off (those MDMs need to be off before forming the set because OPS 0 GPCs will attempt to perform flight-critical I/O with these MDMs as part of the GPC initialization). GPC 5 has its output switch placed in BACKUP (Term B is set to 1), and the RHC engage pushbutton is depressed. The ground will see, via downlist, that GPCs 1-4 receive all engage discretes = 1 and I/O Term B = 1. GPC 5 will see three of three engage discretes = 1 and its I/O Term B = 0. The GPCs are then disengaged using the BFC disengage switch and then reengaged from the other RHC. The BFS is once again disengaged; however, the disengaging is accomplished by a BACKUP to NORMAL transition of GPC 5's output switch.

Since FF 1-3 are powered off, the set will see no flight-critical I/O errors; therefore, the special I/O Error-BFS engage test is never done. After two consecutive cycles of legal engage discretes, the PASS software will respond as follows:

A. Bus masks are set on FC, PL, and BFS controlled DK buses (via the BFC CRT SEL switch)

B. Drive the output talkback of the GPC to barberpole

C. Condition the cyclic monitor software to monitor and respond to a transition from engage to disengage via the BFC disengage switch, or a transition of the output switch from BACKUP to NORMAL (this allows Essential Bus 3AB to set the "select off" discretes in the BFC to the high state, ref. SSSH drawing 8.2a).

Therefore, an engage with no FC I/O errors, such as an on-orbit engage seen by the SM GPC, will result in bus masks being set as well as engaged switch monitoring (as mentioned above). This action is in direct contrast with the PASS GPC's normal reaction of going to software halt upon an engage.

For a real world application of this process, see the PASS RECOVERY AFTER BFS ENGAGE (DPS FRP-4) procedure in the Malfunction Procedures book. The SM normally has its output switch in the TERM position to prevent it from communicating over the flight critical buses. The FRP has the crew take the SM output switch from TERM to NORM, and then reengage the BFS so that the BFC module connected to the SM GPC can latch in the engage discretes. If the I-Fail CAM is lit (which would normally be the case if a dual G2 set was

up at the time of engage), the SM mode switch is cycled from STBY to RUN in order to reset the light. If the I-Fail is not lit (due to a CS split between the SM and a single G2), the moding from STBY to RUN does not harm anything. An OPS XO1 PRO is performed for final recovery of the SM machine.

Upon transition to OPS 0 (STBY-RUN), the primary GPC does not attempt to read the engage discretes before bus reconfiguration. Essentially, this is because the first GPC up (which, after a BFS engage, the SM machine would be) would take control of all four strings. However, no I/O is performed, and no dual commanders would result. Later on during the primary GPC initialization process (a matter of milliseconds), the GPC will examine the engage discretes and set the appropriate masks.

8.7.2 Hand Controller Engage Driver (HCED) Operations

As stated earlier, the HCED receives the engage discretes from the RHCs as shown in figure 8-11. Loss of either power source prevents the HCED from detecting the engage discrete or broadcasting the discrete to the engage logic in the five BFC modules. Refer to figure 8-11 for the DDU and control bus power interfaces.

Troubleshooting failures in the HCED is limited since there are no telemetry data from within the logic. If the logic should happen to fail on, then all BFC modules will receive the false engage discrete and transmit it to their respective interfacing GPC. The IOP status word B in the downlist for each GPC would indicate the discrete to be ON. The same ground indication would be present if the failure was in the RHC or in the discrete driver in the front end of the HCED. However, this failure mode could be isolated by powering down the interfacing DDU and toggling the DISENGAGE switch; i.e., the false engage discrete should go away if the problem was at the DDU. If the engage discrete remains ON with the DDU power OFF, then the failure is in the HCED. No technique is available for detecting failed OFF discretes prior to the actual engage. However, the BFS engage function is performed during the prelaunch checkout for each mission.

8.7.3 BFS Engage Software and Mechanisms

Certain BFS functions are operative only after BFS is engaged. Consequently, when the BFS is in the preengaged state, BFS will not annunciate or display the following items:

A. RCS jet failures:
Fault msg - "F(R) (L) RCS JET"
GNC SYS SUMM 2 - "ON/OFF" in jet fail column

B. OMS gimbal failures:
Fault msg - "R(L) OMS GMBL"
MNVR display - "↓" by failed gimbal

C. OMS chamber pressure failure:
Fault msg - "R(L) OMS PC"
MNVR display - "↓" by failed engine

D. MPS command path failure: Fault msg - "MPS CMD C(L) (R)"

E. Separation inhibit: Fault msg - "ET SEP INH"

F. Aerosurface positions (MM 301-303): GNC SYS SUMM 1 - positions shown may be incorrect

Once every minor cycle (40 ms), the engage and terminate B discretes are tested to determine if the software operation should change from the pre-engage to the engage state. Table 8-VII identifies the discrete configurations and software responses.

TABLE 8-VII.- BFS SOFTWARE ENGAGE RESPONSES

Engage 1,2,3	Terminate B	Software response
2 of 3 = 1	0	Engage
3 of 3 = 1	0	Engage
3 of 3 = 0	1	Preengage
Any other combination		Preengage/log "Illegal Engage/Term B" error code (41 HEX) in GPC Error Log-downlisted

Upon recognition of the engage request, the BFS assumes control of the flight-critical buses, inputting sensor data, etc.; attaches guidance and Digital Autopilot (DAP) tasks; assumes control of both BFS selected CRTs; assumes control of the payload buses if not previously commanded; and illuminates the BFC light (panels F2 and F4). Also, the GPC OUTPUT talkback indicator transitions to gray indicating that the BFS is commanding on the flight-critical buses.

If the BFS is subjected to a HALT-TO-STBY or RUN transition with the BFS engage discretes and I/O term B discrete set to the engage condition, the BFS will first perform a preengage startup sequence causing "BCE STRG X PASS" fault messages to be annunciated prior to performing an engage transition.

Currently, BFS does not provide a proper engage transition initialization during the OMS/RCS interconnect process (OMS propellant burn via the 24 aft RCS jets). If BFS is engaged while the propellant burn is in progress, all 24 RCS jets will be terminated without the staggered shutdown. This will result in a pressure surge which may rupture the propellant lines. Therefore, BFS should not be engaged while the OMS/RCS interconnect propellant burn is in progress.

8.7.4 Control Bus Down Impacts

The routing of control bus power from the HCED logic into the engage logic to power the optical isolators (fig. 8-10) causes nonengagable GPCs when CNTL AB3 or CNTL CA1 is down. With either control bus down, all three HCEDs associated with one RHC will be no-go for engage, which is not a major impact since the other RHC/HCEDs/control bus are still operational. However, the control bus being down does take away the necessary power from the optical isolators in the engage logic. This prevents the engage discretes from the HCEDs powered by the good control bus from getting to the engage latches. Consequently, the GPCs interfacing with these modules are unaware of the BFS engage and will continue to operate. The result is that one or more PASS GPCs will need to be taken to halt manually, and, depending on where the BFS software is resident, the BFS may be no-go for engage. If time permits following a control bus down, the BFS may need to be moved to a GPC which can be engaged. Note that the BFS assumes control of all flight-critical buses and a second CRT when engaged, and may bypass input data because of dual commanders; i.e., both PASS and BFS commanding on the same flight-critical data bus, causing I/O errors. Once the remaining PASS GPCs have been taken manually to halt, a BFS I/O RESET may need to be performed to recover bypassed inputs, and Flight Control system channels reset. Table 8-VIII highlights the control bus down engage capabilities.

TABLE 8-VIII.- CONTROL BUS DOWN ENGAGE CAPABILITIES[7]

Control bus down	Engage capability status			
	L RHC	R RHC	BFC	GPC
CNTL AB3	No engage	Engage	1A No-go	GPC 1 - No engage
	No engage	Engage	1B No-go	GPC 4 - No engage
	No engage	Engage	2A No-go	GPC 2 - No engage
	No engage	Engage	2B Go	GPC 5 - Engage
	No engage	Engage	3A Go	GPC 3 - Engage
	No engage	Engage	3B Go	N/A
CNTL CA1	Engage	No engage	1A Go	GPC 1 - Engage
	Engage	No engage	1B Go	GPC 4 - Engage
	Engage	No engage	2A Go	GPC 2 - Engage
	Engage	No engage	2B No-go	GPC 5 - No engage
	Engage	No engage	3A No-go	GPC 3 - No engage
	Engage	No engage	3B No-go	N/A

8.7.5 DDU +15 V dc Power Loss Impacts[7]

Loss of DDU +15 V dc power supply A, B, and/or C affects the RHC's ability to issue the engage A, B, and/or C discretes, respectively. This loss also disables the interfacing HCEDs from receiving the discretes. It takes only one failure of the three internal DDU power supplies to declare the RHC no-go for engage; i.e., three of three engage discretes from an RHC are required. Note that the telemetry DDU power supply failure indication can be misleading in that the +28 V dc can be failed; however, the +15 V dc is good.

8.7.6 RPCA Power Loss Impacts[7]

Loss of power from the RPCA not only disables the BFC module's engage logic, but also fails the associated GPC, making the engage logic nonapplicable. The HCED will not be affected and will continue to support the other BFC modules.

8.8 DISENGAGE OPERATIONS[4]

When the BFC DISENGAGE switch is switched to the momentary DISENGAGE position, all BFC modules unlatch (reset) their engage latches. This forces the engage logic to return to the preengage state depending on the GPC OUTPUT switch. The logic will then stay in the preengage state until another engage is detected.

Certain BFS functions, applicable only to the engaged BFS, are not detached following a PASS re-IPL and subsequent BFS disengage. The result can be the erroneous annunciation of certain command-related failure indications after a PASS reengage or BFS disengage. For example, RCS jets can be declared failed on or failed off. Once the BFS is moded to OPS 0, the annunciation is halted. Therefore, subsequent to a BFS disengage, ignore BFS GNC command-related fault annunciation.

If BFS is moded from RUN to STBY while the "BFC" engage lights are flashing, the engage lights will change to either always on or always off if a message reset is not done prior to moding the GPC to STBY. If BFS transitions from engaged to disengaged while the GPC mode switch is in STBY, the "BFC" engage lights will remain on. This situation can be rectified by putting the GPC Mode switch in RUN and disengaging the BFS. A message reset may need to be performed in order to clear the engage lights.

8.8.1 BFS Software Response to a Disengage

When the BFS software recognizes all three engage discretes change to the logic "0" state and the terminate discrete change to the logic "1" state, the BFS ceases commanding on the flight-critical buses, ceases commanding on the second CRT bus associated with the appropriate BFS CRT select discretes, and ceases commanding on the payload buses if the BFS MODE switch is in

STANDBY. Also, the GPC OUTPUT talkback changes to barberpole, and the BFC lights are turned off (if BFS maintains commanding on the payload buses).

8.8.2 PASS Disengage Software

Prior to BFS disengage, the PASS computers are recovered (IPL) in OPS 0. Since the engage/terminate discretes are set to logic "1", the PASS OPS 0 software executes with the flight-critical and payload bus activity halted and bus masked, and the appropriate CRT bus activity halted and bus masked according to the BFS CRT select discretes. When the PASS software recognizes that three of three engage discretes change to the logic "0" state and the terminate discrete changes to the logic "0" state, the PASS assumes control of the flight-critical buses immediately and the CRT bus after three major cycles, which is 2.88 seconds. The CRT commanded is the second CRT of the pair as determined from the BFS CRT select discretes. Commanding will also resume on the payload buses if the BFS GPC MODE switch is in STANDBY and BFS disengaged.

8.8.3 Control Bus Down Impacts

When either CNTL AB3 or CNTL CA1 is down, up to two engage latches in each module cannot be unlatched by the BFC DISENGAGE switch (fig. 8-19) and will continue to cause engage discrete(s) to be issued to the interfacing GPCs. Note that the terminate B discrete will change state if one of the engage discretes is taken away. Since at least one engage discrete will be present, neither the PASS nor BFS disengage software will be satisfied and perform the disengage function; i.e., all three engage discretes must change to logic "0". If the above scenario takes place, neither the BFS or PASS will be commanding the flight-critical buses--no control.

For the no-control bus down situation, the PASS Recovery After BFS Engage Malfunction procedure (FRP-4) re-IPLs the PASS computers without cycling GPC power (maintains the BFC output discretes). The maintained engage/terminate output discretes keep the re-IPL'd GPCs from commanding on the flight-critical buses while the BFS is still engaged. However, for the control bus done case, the only way to unlatch all engage latches is to cycle GPC power. For example, the PASS GPCs can be re-IPL'd without cycling power (nominal FRP-4 procedure) and then powered off in HALT. The BFS GPC can then be powered off (ceases commanding on the flight-critical buses) and then the re-IPL'd PASS computers can be powered up, moded to RUN (first PASS GPC assumes commanding on the flight-critical buses) and then transitions into the desired software configuration.

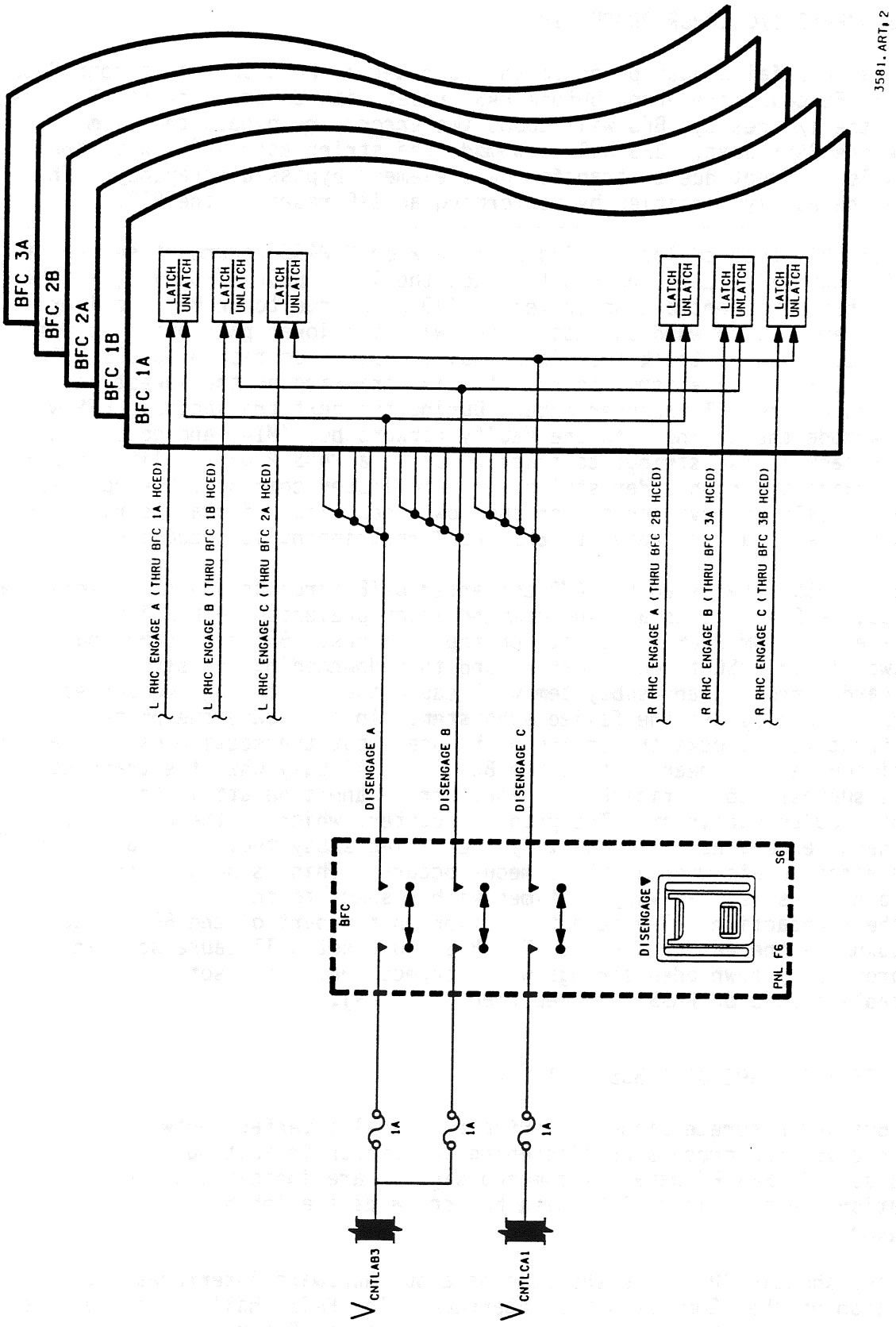


Figure 8-19.- BFC DISENGAGE switch/control bus interface.

8.9 BFS/PASS I/O ERROR PROCESSING

A. If an LRU fails just prior to an I/O reset in PASS such that both PASS and BFS count one error before PASS resets its error counters (because of the I/O reset), BFS will count two errors and bypass the element before PASS does. BFS will downmode the string associated with the failed element due to transfer data element bypass discrepancy. This can be easily rectified by performing an I/O reset to the BFS.

B. With this type of logic being used, a 2 on 2 PASS redundant set split will normally cause the BFS to track the lower pair. However, if the BFS has experienced a nonuniversal I/O error due to an MIA receiver problem on a forward bus associated with the lower pair prior to the split, BFS will track the higher pair. Upon execution of a BFS I/O reset, BFS will attempt to reinitialize tracking on the lowest pair based on the AFT transfer data. During the next transaction, BFS will downmode the string with the faulty forward bus (MIA) and go standalone for lack of two strings to track. During a PASS 2 on 2 split, if BFS is tracking the high order string pair and listen command I/O error was previously observed on one of the low order flight forward strings, do not attempt an I/O reset to BFS until restringing is accomplished.

C. When PASS detects an MDM A/D converter BITE error on any flight-critical bus, PASS initiates a no-go command which prevents the BFS from receiving MDM PROM input data on the same bus. BFS reacts by logging two listen (MSC timeout) errors and then downmoding the string. A "hard" error on any subsystem will cause the BFS to log two consecutive errors and bypass the failed subsystem. In this case the primary effectively blocks the entire high rate input transaction causing an MSC timeout error (meaning that the BCE is still busy when the transaction is supposed to be finished). That error cannot be attributed to any particular subsystem. The program counter, which in the case of a "hard" error, helps to identify the failed subsystem, is an unreliable diagnostic aid when an MSC timeout occurs. This is because the program counter is read at varying times with respect to the scheduled end of the transaction. The safest reaction on the part of the BFS is to downmode the entire string. The BFS I/O reset will cause upmoding of previously downmoded strings with correct indicators set in the bypassed table (based on PASS-provided transfer data).

8.10 BFS/PASS GPC DATA BUS CONTROL[1]

Data Bus and Discrete Signal Interface: signal interface between PASS and BFS involves the process of "listening to" and/or inhibiting conflicting signals. All BFS FC data bus command signals are inhibited during PASS operation. Also, all PASS FC data bus commands are inhibited upon BFS engagement.

All five shuttle GPCs have the same data bus hardware interfaces with the exception of the PCMMU data bus interface. The PASS shall support data bus controls across OPS transitions when the BFS is in RUN mode.

The following gives criteria needed for BFS and PASS to control the affected data bus:

A. Flight Critical Buses:

Control capability is determined by hardware response to manually set switches. Acknowledgment of control commands is by software response to computer discrete inputs.

BFS GPC Control -- On BFC Engage
PASS GPC Control -- On BFC Disengage

B. Payload Buses:

Control is determined by software response to the BFS GPC RUN discrete and the BFS RUN intercomputer discrete.

BFS GPC Control -- BFS RUN discrete = 1 (switch in RUN). PASS will relinquish control of the P/L buses within two major cycles of the setting of the BFS RUN discrete to a "1" by halting the BCEs and by setting the appropriate bus masks.

PASS Control -- BFS RUN discrete = 0. The BFS GPC will relinquish control of the P/L buses prior to setting the BFS RUN discrete to "0". PASS will assume control of the P/L buses if the BFS RUN discrete is "0" by resetting the appropriate bus masks and restarting the BCEs when required.

C. Instrumentation PCM (IP) Buses:

PCM toggle buffer 5 will be assigned to the BFS GPC independent of the location of that GPC.

D. Intercomputer Communication Data Buses (ICC):

All shuttle GPCs have a software dedicated ICC data bus. ICC data bus message transactions are only applicable to the PASS; the BFS does not contain any software code for transmission of ICC bus messages, and all five ICC BCEs shall be maintained in HALT in the BFS. The PASS keeps the BCE of the BFS-controlled ICC data bus in the HALT state.

E. Display Data (DK) Buses:

The BFS will be initialized at System Reset (transition from HALT) to relinquish control of DK4. The forward CRT assignments to BFS are controlled by the BFC CRT Display switch setting. Upon engage, the BFS will assume control of the two CRTs that are reflected in the switch position; in the preengage or disengage state, BFS will only maintain control of one CRT (the first CRT number in the current switch position).

PASS will relinquish control of the data bus within 200 ms after BFS engage. Following BFS IPL, PASS may assume control of the DK buses not assigned to the BFS.

8.11 LISTEN COMMAND PROCESSING[1,4]

The BFS, while in the preengage state, must listen to PASS transfer data containing PASS operation status (e.g., element bypasses). The BFS must also acquire sensor data by listening to data being retrieved by the PASS computers. In order not to be polluted with bad data, the BFS downmodes any string where non-nominal PASS computer operations are detected. The string downmode criteria are:

A. BFS fails to receive listen commands from a PASS computer for two consecutive minor cycles on a flight-critical (FC) bus for a given input transaction; e.g., HFE input transaction FF2 LSN CMDs are not received on two consecutive minor cycles,

B. A string's copy of 25 Hz transfer (XFER) data received from PASS (on FC 5,6,7 and 8 buses) does not match the other copies; e.g., bit by bit comparison of string 1 XFER data does not compare with string 2,3, and 4 copies,

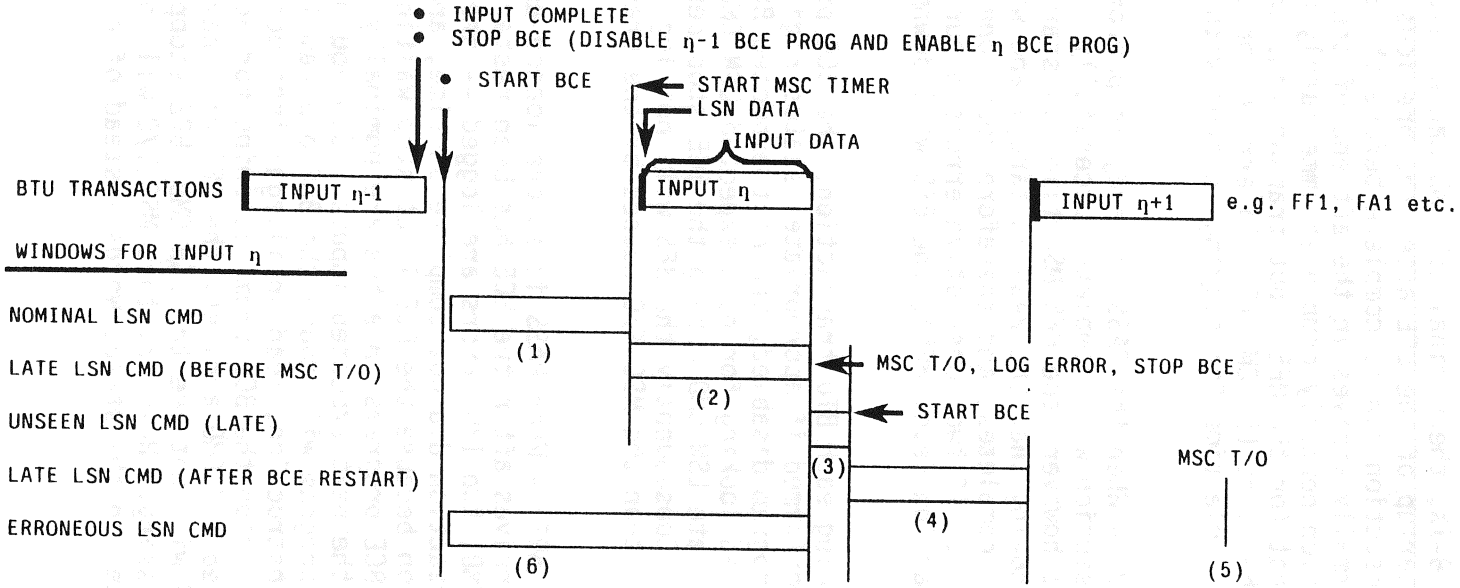
C. The element bypass indicators received in the XFER data shows that the PASS computers bypassed a data element in an input transaction when BFS detected no input errors with the element; e.g., PASS bypasses IMU 2 when BFS detects no IMU 2 I/O Error, and

D. Less than two strings are operating nominally; e.g., BFS is tracking strings 1 and 2 and string 1 LSN CMDs cease - BFS downmodes both strings.

All input transactions of FC MDM, EIU, NSP, and MTU data initiated by PASS are preceded by LSN CMDs containing information (BCE program identification index) the BFS requires to start the correct BCE program for receiving the incoming data. The BFS, in order not to be fooled, performs several LSN CMD checks to make sure the PASS computer commanding the bus is operating as expected. These checks involve testing to see if the LSN CMD is the one expected and if the LSN CMD arrived when expected. Table 8-IX highlights the various LSN CMD windows and the following paragraphed discuss the BFS response.

Flight-critical input transactions (i.e., HFE, MFE, NSP, and MTU) are composed of one to four BTU transactions. For example, the HFE input transaction on the FC bus 8 contains three EIU BTU transactions and one FA4 BTU transaction. Each BTU transaction is composed of one or more input elements; e.g., FA4 return word, R OMS Pc, and Prom Seq 3-10 (D). Also, three XFER data transactions from PASS to BFS occur only on the aft FC buses; i.e., 25 Hz PASS element bypass status (every minor cycle), 1 Hz PASS state vector data (minor cycle 15) and 1 Hz PASS calibration data (minor

TABLE 8-IX.- BFS LISTEN COMMAND WINDOWS (TYPICAL)



Notes:

(1) LSN CMD received early enough for data to complete before MSC timer expires

(2) LSN CMD received; however, data does not complete before MSC T/O;i.e., MSC T/O logged against input η

(3) LSN CMD arrives after MSC T/O (while BCE is momentarily stopped) and is not seen, i.e., MSC T/O logged against input η

(4) LSN CMD arrives after MSC T/O and BCE reinitialized looking for next LSN CMD (η+1). BCE branches to illegal instruction and stops causing next LSN CMD to be missed. MSC T/O logged against input η.

(5) LSN CMD/inputs (η+1) are not seen because BCE is stopped. MSC T/O is logged, with illegal instruction in status word, against input η+1.

(6) BCE stops on illegal instruction. MSC timeout is logged with illegal instruction against input η.

18820*053

cycle 23). Each of these FC BTU and XFER transactions is preceded with an LSN CMD to complete the LSN CMD prior to each DEU poll response, which cues the BFS to start the DK bus Poll Response BCE program.

Nominal LSN CMD - The BFS does not specifically verify that the LSN CMD is received at the correct time, but indirectly makes the verification by checking to see if the BTU transaction completes before the MSC timer expires. As shown in table 8-IX, the nominal LSN CMD can arrive as early as immediately following the startup of the BCE after the previous transaction, and also the entire BTU transaction can be complete before the BFS MSC timer is started. The LSN CMD normally arrives in the aft end of the nominal LSN CMD window, and the transaction normally completes immediately prior to the MSC timer expiring. Note that for the HFE input transaction, the BFS specifically determines when the first LSN CMD (EIU 1) arrives on the aft FC buses and adjusts the minor cycle time for staying in sync with the earliest PASS computer.

LATE LSN CMD (BEFORE MSC T/O) - When the PASS is tardy in performing an input transaction, the transaction will complete successfully in PASS (MSC timer is also started late); however, the BFS MSC timer is started at the normal expected time (not knowing that the PASS is tardy) and will time out before the input transaction completes, and therefore count one error against the input transaction. If two consecutive errors occur (e.g. FF2 LSN CMDs) on an input transaction (e.g. HFE), the BFS will downmode the entire string; e.g. string 2.

UNSEEN LATE LSN CMD - Following each BTU transaction, the BCE program is stopped, error logging is performed if appropriate, status registers cleared, the previous BCE program disabled, the next expected BCE program enabled, and the BCE restarted looking for next expected LSN CMD. If PASS should happen to transmit a late LSN CMD during the BCE stop period, the BFS will never see the command. Consequently the BFS will not log any additional error condition other than the MSC T/O caused by the transaction not completing on time.

LATE LSN CMD (AFTER BCE RESTART) - When PASS is so overloaded and late that the last expected LSN CMD arrives after the BCE has been restarted looking for the next expected LSN CMD, two I/O errors are logged -- one MSC T/O because the last input transaction did not complete on time, and one MSC T/O on the next input transaction because the BCE is at stop waiting for error processing. Note that all BCE programs on a bus are nominally disabled except the BCE program for the next expected inputs. Consequently, a late LSN CMD for a disabled BCE program will cause the BCE program to branch on the disable to an illegal instruction, load an illegal instruction indictor in the status registers, and stop the BCE from listening for additional LSN CMDs. When the next expected LSN CMD arrives, the BFS will have started the MSC timer; however, the BFS will not see the LSN CMD (BCE stopped) and will perform MSC T/O error processing. However, this MSC T/O will have an illegal instruction indicated in the status words instead of all zeros.

ERRONEOUS LSN CMD - Lastly, if an erroneous LSN CMD (not previous late LSN CMD) is received, the BCE again is stopped causing the BFS MSC timer to expire without inputs complete, at which time an MSC T/O is logged with illegal instruction indicated in the status words; i.e., similar to a late LSN CMD.

The requirements are not clear what to do when an unexpected (erroneous or late) listen command is received. Initially, the BFS designers downmoded the string if only one unexpected LSN CMD was received; however, they had second thoughts that this was too severe. They finally decided to branch to an illegal instruction and cause BCE to stop. This indirectly causes the next input transaction to be missed and an MSC T/O to be logged with illegal instruction indicated in the status words. This is reasonable except when PASS gets overloaded and late in transmitting LSN CMDs. The right time skew among the PASS computers and the total "lateness" can cause BFS to downmode one or more strings. An I/O reset will regain the string(s); however, some crewmembers consider this a nuisance. The only fix is to tighten up the PASS timing or possibly restart the BFS BCEs immediately after unexpected listen commands.

8.12 HIP POCKET BFS-RELATED IFM PROCEDURES AND PERIPHERAL DATA

As mentioned previously in section 8.7.1, methods exist which will enable the crew/ground to engage the BFS without taking the PASS to the WAIT state. Two additional methods are presented here:

A. BFS Engage While Retaining PASS

This method deals with string deassignment. This procedure was verified in SAIL on 8-5-81:

- ✓ BFS GPC OUTPUT - BACKUP
- BFS GPC MODE - RUN (so that the GPC can turn off BFC at end)
- BFC CRT DISP SEL - ON (3 + 1)
- SM GPC OUTPUT - NORM
- Deassign FC strings from GNC (✓ BFC LT - OFF)
- RJDs - OFF
- BFS engage
- ✓ ASAP - RJDs ON

  Backout:
  - Set up GNC NBAT for strings
  - All RJDs - OFF
  - BFS DISENGAGE
  - GNC, OPS 301 PRO
  - ASAP - RJDs ON
  - BFS I/O RESET (✓ BFC LT - OFF)
  - BFS GPC OUTPUT - NORM
    MODE - STBY
  - SM GPC OUTPUT - TERM
  - MTU TRY

Similar to the above option, this method takes the GNC GPCs OUTPUT switches to TERM and prevents communication over the FCs (thus preventing I/O errors).

- ✓ BFS GPC OUTPUT - BACKUP
- BFS GPC MODE - RUN (so that the GPC can turn off BFC at end)
- BFC CRT DISP SEL - ON (3 + 1)
- GPC/CRT 01 EXEC (prevent dual commanders)
- GNC GPC(s) OUTPUT - TERM
- RJDs - OFF
- BFS Engage
- ASAP - RJDs ON

  Backout:
  - All RJDs - OFF
  - BFS DISENGAGE
  - GNC GPCs OUTPUT - NORM
  - GNC I/O RESET
  - ASAP - RJDs ON
  - BFS I/O RESET (✓ BFC LT - OFF)
  - BFS GPC OUTPUT - NORM
    MODE - STBY
  - GPC/CRT 11 EXEC
  - BFS GPC MODE - STBY

B. BFS Engage for Hand Controller Discrete Fail

Noting that three of three engage discretes are necessary for the PASS to recognize an engage, the PASS will not recognize an engage if one of the hand controller engage discretes is failed. However, it is still possible to engage the BFS. In order to do so, the crew must perform the following steps:

1. Put the BFS GPC OUTPUT switch in NORM (in order to set its I/O Term B to 0)

2. Take one of the PASS GPC's (1-4) OUTPUT switch to BACKUP (in order to latch in the BFS SEL to 1 for the entire GPC configuration within the BFC)

3. Depress the PB for engage (the BFS only needs two of three engage discretes set high, an I/O Term B set to 0, and the BFS SEL discrete set to 1 for an engage condition)

4. If BFS properly engages, perform this step; if the BFS does not engage, do not perform this step because the PASS set will be gone with no BFS available. Take the MODE switches for GPCs 1-4 to HALT (the reason: PASS has to be manually taken to halt because PASS will not go to S/W halt automatically due to the failed discrete. Also, because power is still being supplied to the GPC and the BFC modules, the BFS SEL discrete is still set -- it does not matter which GPC set the discrete within the BFC; the only requirement is

that at least one of the GPCs sets the discrete so that both PASS and BFS know that some GPC has been selected as the BFS GPC.)

C. BFS I/O Term B Recovery Procedures[7]

1. Loss of essential bus ESS 3AB power to the GPC OUTPUT switches is one of many single point failures that prevent the BFS from being engaged. This procedure provides a troubleshooting outline that integrates together both failure identification and IFM procedural activities.

   Loss of the BFS I/O Term B discrete can be detected by monitoring the BFS Downlist. The failure is transparent to the crew since the OUTPUT talkback (tb) is slaved to barberpole (FC transmitters inhibited) while the BFS is in the preengage state.

   Troubleshooting requires an SM GPC to isolate whether the failure is BFS GPC, BFC, or GPC OUTPUT switch related, or something that affects all computers. If the failure is also SM GPC related, Panel (PNL) O6 must be opened and ESS 3AB fuse 49 inspected. Opening PNL O6 is risky since it contains POWER, MODE, OUTPUT, etc. switches for all the computers, and flexing the wiring bundles can possibly cause unknown problems. In other words, the CHIT requesting concurrence to open PNL O6 will probably receive much attention from the technical and managerial communities.

   Whether the BFS capability can be recovered depends on the failure location. The possibilities are highlighted as follows. The failure location assumes that the BFS is located in GPC 5:

   - GPC 5 OUTPUT sw BACKUP position open,
   - GPC 5 OUTPUT sw SEL wire open to BFC 2B,
   - GPC 5 Discrete 13 complement wire short-to-ground or true wire open,
   - BFC module 2B I/O Terminate B logic inverted,
   - Open wire between ESS 3AB and GPC 5 OUTPUT sw,
   - Short-to-ground between fuse and GPC 5 OUTPUT sw,
   - Short-to-ground between GPC 5 OUTPUT (BACKUP) sw and BFC 2B,
   - Transient short-to-ground between fuse and BFC 2B.

   Figure 8-20 summarizes these failures pictorially, and the trouble-shooting procedural outline is presented in figure 8-21. Each procedural block is numbered in figure 8-21 with corresponding failure indicated on figure 8-20.

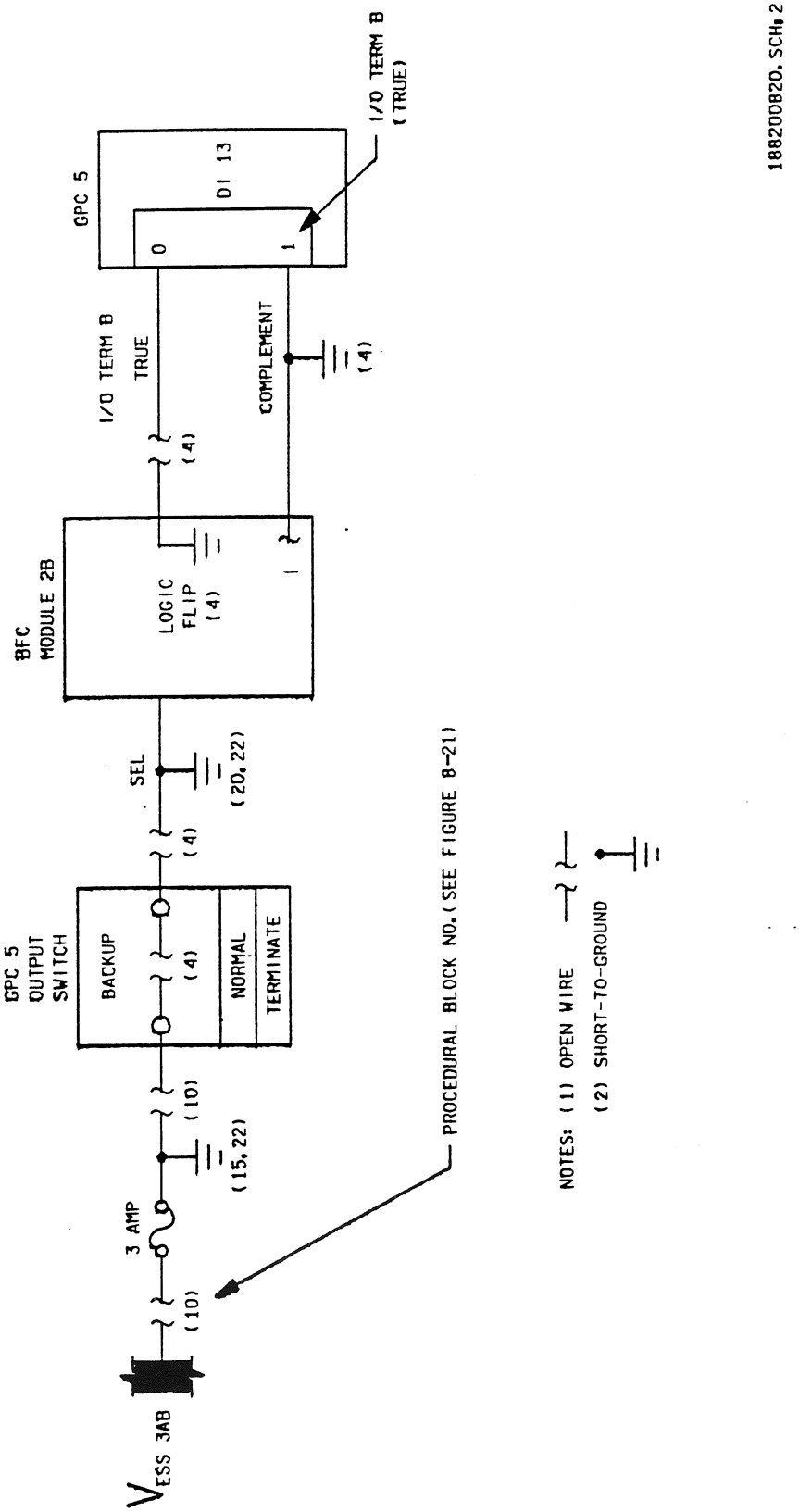


Figure 8-20.- BFC I/O TERMINATE B loss pictorial.

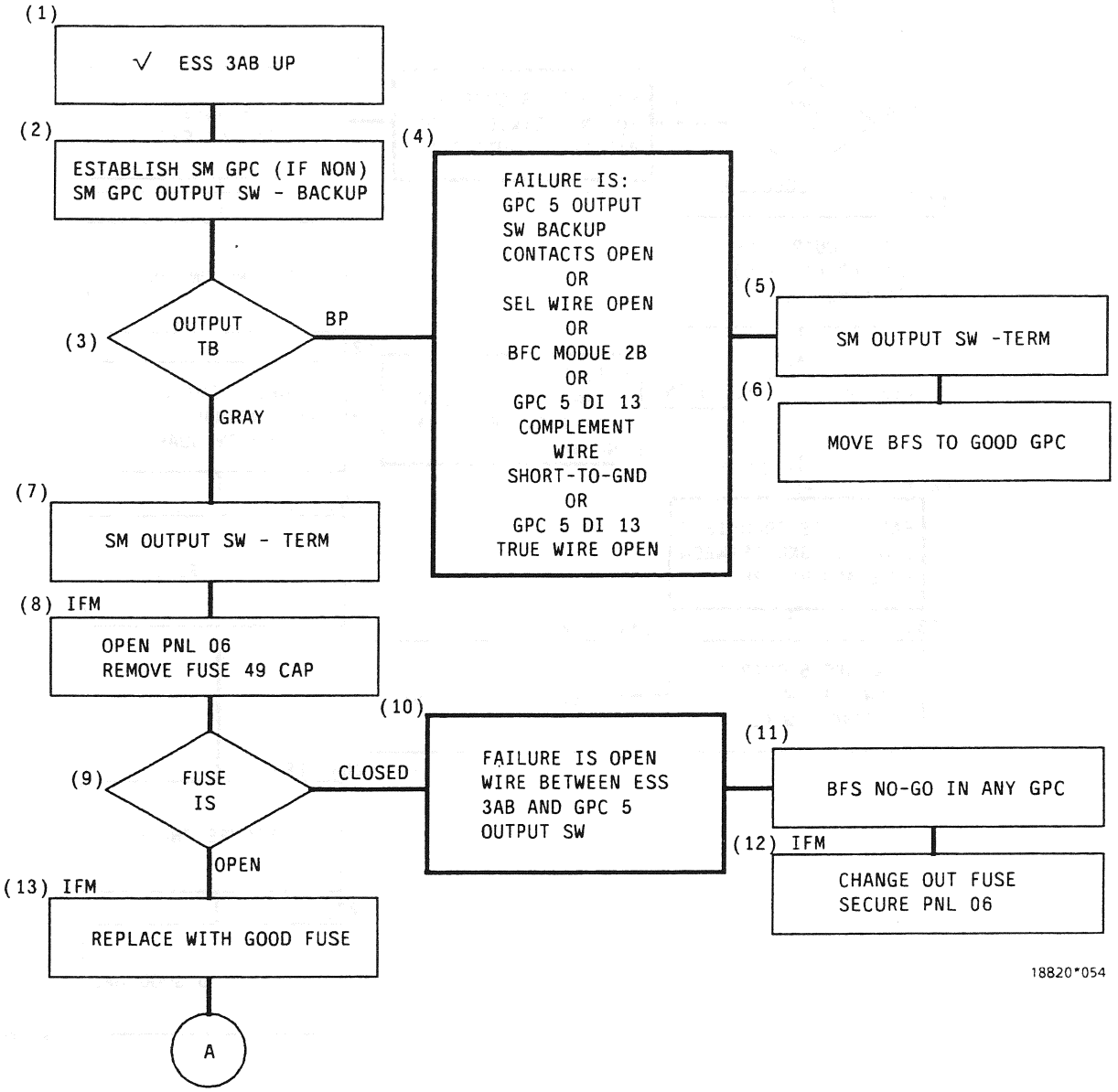


Figure 8-21.- BFS I/O term B loss (procedural outline).

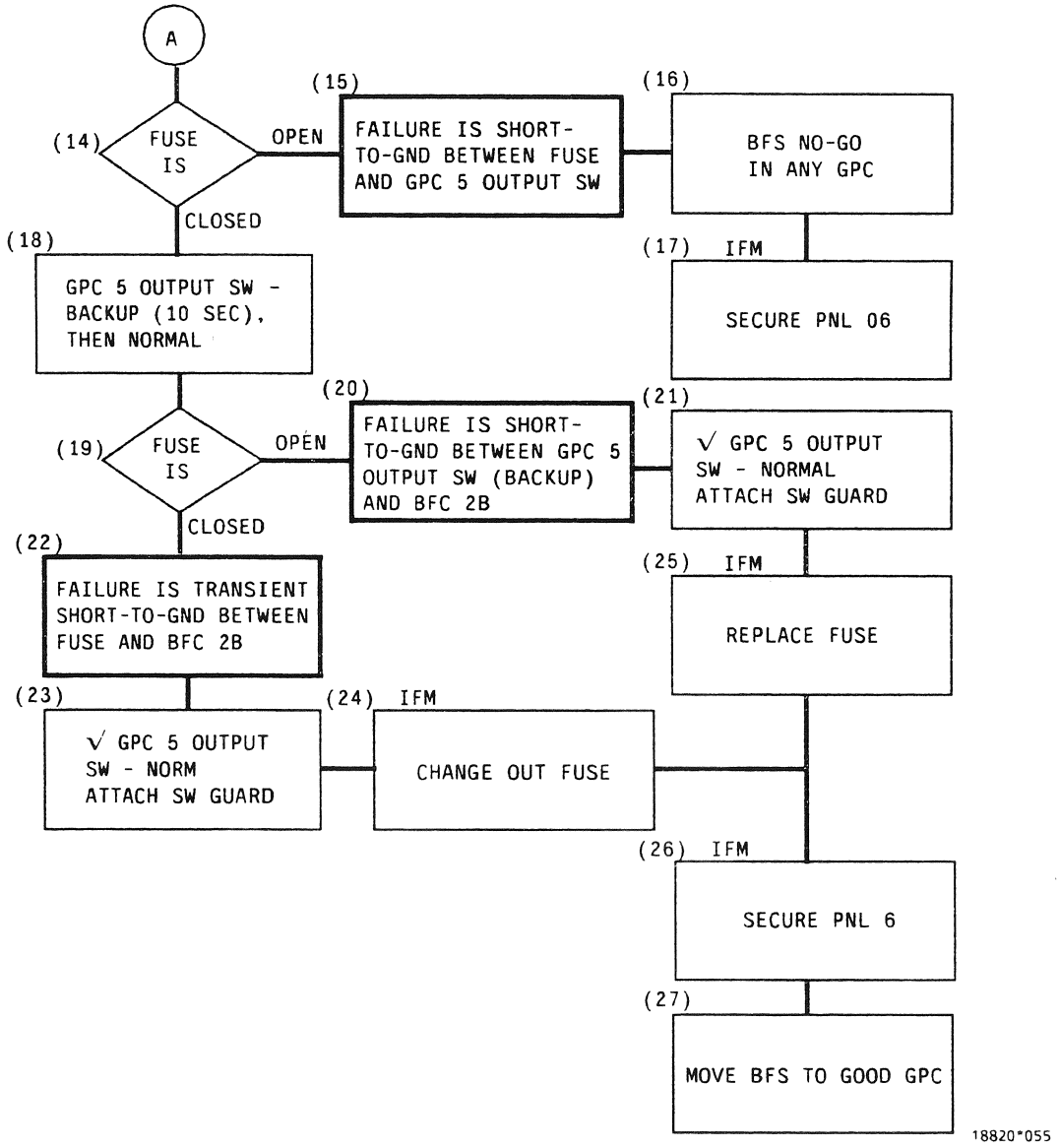


Figure 8-21.- Concluded.

Block 1 - Verify that ESS 3AB is currently up. The flight controller should have already verified that ESS 3AB was not down during the I/O Term B loss period.

Block 2 - If the failure occurred during ascent, this troubleshooting procedure would probably be performed post-DPS reconfiguration and an SM GPC would exist. If the failure occurs predeorbit burn, an SM GPC would need to be established. However, if bad weather is closing in on the landing site, the Flight Director may choose to immediately move the BFS to another computer and gamble that the failure is related only to GPC 5.

Block 3 - Unlike the BFS, the PASS system software drives the OUTPUT talkback to barberpole (bp) when the I/O Term B discrete is true. Consequently, with the SM GPC OUTPUT switch in BACKUP and the OUTPUT tb indicating bp, the determination can be made that the failure is only BFS related and that the 3 amp fuse is closed.

Block 4 - The GPC 5 OUTPUT (BACKUP position) switch is either (1) not supplying ESS 3AB power to the SEL signal (open circuit), (2) the SEL signal wire is open, (3) the BFC 2B module has an electronic failure, (4) the TERM B wire (complement side) has a short-to-ground, or (5) GPC 5 Discrete 13's true side is open. Note: If the failure is a BFC electronic component inverted and the BFS was to be engaged, there is no assurance that the I/O Term B will remain in the false state.

Block 5 - Reconfigure SM GPC switch to nominal.

Block 6 - Move BFS to a good GPC, usually GPC 2.

Block 7 - Reconfigure SM GPC switch configuration to nominal.

Block 8 - PNL O6 is held in place by 16 screws and is hinged on the right side. The fuse is covered with a cap that is safety wired. The fuse is rated to 3 amps, plugs in via two prongs and can be inspected through a window on top.

Block 9 - Inspect fuse circuit through window (or test with ohms meter).

Block 10 - Since neither GPC 5 or the SM GPC received the I/O Term B signal, and the fuse is closed, an open circuit exists between the ESS 3AB bus and the GPC 5 OUTPUT switch, preventing the SEL signals from going high (28 V dc) to the respective BFC modules.

Block 11 - Since the failure has been determined to be multicomputer related (GPC 5 and SM GPC), the BFS will not engage in any computer. BFS can still perform the BFS SM function.

Block 12 - Change out fuse in case there is a hidden open circuit in the fuse and resecure PNL O6.

Block 13 - Replace the open fuse with a good fuse.

Block 14 - Inspect fuse circuit through window.

Block 15 - Since the fuse is again open, a short-to-ground exists between the fuse and the GPC 5 OUTPUT switch.

Block 16 - Since the open fuse prevents ESS 3AB power to any GPC OUTPUT switch, the BFS is no-go for engage in any GPC.

Block 17 - Resecure PNL O6.

Block 18 - Since the fuse was found open in Block 9, and it remained closed when replaced, a short-to-ground may exist on the GPC 5 OUTPUT switch to BFC 2B SEL signal wire. Momentarily positioning the switch to BACKUP will supply ESS 3AB power to the SEL signal wire (or to the possible short-to-ground which would cause the fuse to open).

Block 19 - Inspect the fuse circuit through the window.

Block 20 - A short-to-ground exists on the GPC 5 OUTPUT switch sel signal wire; i.e., the short is only present when the GPC 5 OUTPUT switch is in the BACKUP position.

Block 21 - Safety guard switch in NORMAL position.

Block 22 - If the fuse now remains closed, a transient short-to-ground must have occurred between the fuse and the GPC OUTPUT switch or between the GPC OUTPUT switch and the BFC 2B module on the SEL signal wire.

Block 23 - Safety guard switch in NORMAL position in case the transient short-to-ground occurred on the SEL signal wire; i.e., do not use the GPC 5 OUTPUT BACKUP position.

Block 24 - Change out fuse in case there is a hidden open circuit in the fuse.

Block 25 - Replace the fuse.

Block 26 - Secure the PNL O6.

Block 27 - Move the BFS to a good GPC.

Lastly, the IFM procedure for opening/closing PNL O6 has been performed (6/11/83) without difficulty on the SAIL crew station by T. Pierson. The station was not powered up and the fuse cap was not removed, nor was the fuse inspected. R. Robbins verified that in the past PNL O6 has been opened at KSC; however, the computers were never powered up at the time.

27. A power cycle (power drop) to the BFC may cause the BFS's I/O Term B to be "turned off" or set low. This would be seen on the ground, provided a valid downlist exists; however, the crew would not be able to detect this. The significance of this problem lies in the situation that may cause the power drop.

    If the power drop is the result of a lightning strike, it would be highly likely that the GPCs would also suffer from the same problem. In this case the PASS computers would fail, but the BFS would initiate a power on restart. The crew would then engage the BFS upon loss of control of the vehicle. However, the BFS may not engage due to the improper setting if the I/O Term B. If this occurs and no additional damage has been done, the solution to the problem is to cycle the BFS OUTPUT switch to NORM and then back to BACKUP thus resetting the I/O Term B.

    The problem, it it occurs, would be at the delay circuitry found between the BFS select input and the I/O Term B output of the BFC. The attached drawing shows the BFC circuit in question for one GPC. The BFS SELECT will drive the transistor in the optical isolator circuit "on", providing a ground potential (logic 0) at RESET of the D flip flop shown in the delay circuit. At the same time, the signal is inverted at the NAND gate providing a logic 1 at the input of the D flip flop. When this signal appears, the one-shot delay trigger IC will wait for a predetermined length of time (1 sec) and then send a clocking pulse to the flip flop. This pulse provides a logic 1 at the input of the EXCLUSIVE OR (EOR) gate. When the (AND'ed) engage discretes are low (logic 0), the output of the flip flop (Q) will be reflected at the output of the EOR gate which results in the I/O Term B signal to the selected BFS computer.

    When the (AND'ed) engage discretes signal is set high, EOR gate will cause the I/O Term B to be set low; thus, the BFS will now be able to be engaged. If the GPC select is low (as it will be for the PASS computers), the ENGAGE DISCRETES signal will pass through the EOR gate. The I/O Term B will now be set high, thus preventing the GPC from communicating over the flight-critical buses. It is possible for the selected BFS's GPC SELECT (Q) to be low following a power drop given the right conditions of the momentary power drop. If the D flip flop momentarily loses its voltage (V_{CC}), or if V_{CC} drops below 2 volts, the output (Q) will become indeterminate and may not reflect the input at D until the next clock pulse from the ONE SHOT circuit. The I/O Term B may now be low even though the BFS SELECT is high. Please see figure 8-22 for a graphic illustration.

While the above scenario is only a remote possibility, it is nevertheless possible. There are two cases that warrant consideration. The first one deals with an unknown reason the I/O Term B is lost for the BFS. The crew should take the OUTPUT switch to NORM, and then back to the BACKUP position; this may recover the I/O Term B. The second case deals with the loss of I/O Term B in conjunction with a massive power cycle or drop that may result in the PASS computers going away; the crew should be quickly reminded to cycle the output switch.

8.13 REFERENCES

1. PASS and BFS Software Interface Document (ICD), Rev E, ICD 3-0068-03, 1-9-87.

2. Backup Flight System Program Requirements Document (BFS PRD), Rev E, Doc No. MG038100, 10-14-88.

3. Backup System Services Program Requirements Document (BSS PRD), Rev L, Seq 4, Doc No. MG038101, 11-3-89.

4. Backup System Services Program Description Document (BSS PDD), Change 4, STS 83--0443, September 83.

5. Shuttle Operational Data Book (SODB), Rev D, Vol I, Doc No. JSC-08934, Oct 1984, pp. 4.5.1-9 and 4.5.1-32.

6. Backup Flight Controller (BFC) Interface Schematic, Rev H, Drawing No. VS70-720262, 5-21-85.

7. Failure Modes and Effects Analysis (FMEA) for BFC, BFS-021-0001, FMEA No. 05-8, 2-9-87.

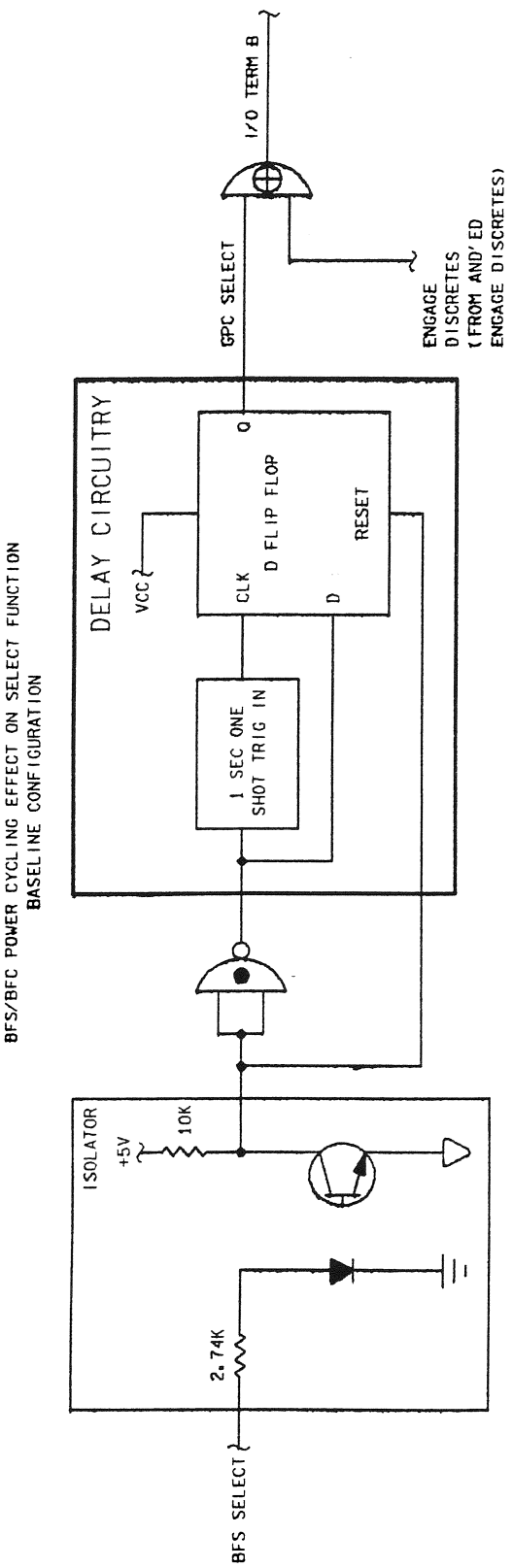


Figure 8-22.- BFS select discrete diagram.

SECTION 9
IPL - AP101 B

CONTENTS

Section		Page
9.1	OVERVIEW	9-1
9.2	HARDWARE PHASE	9-1
9.2.1	Introduction	9-1
9.2.2	Hardware Response Phase	9-1
9.2.3	IOP Microcode Phase	9-3
9.3	SOFTWARE PHASE	9-4
9.3.1	Introduction	9-4
9.3.2	Bootstrap Loader	9-4
9.3.3	GPCIPL	9-4
9.3.3.1	Introduction	9-4
9.3.3.2	Self-Test	9-5
9.3.3.3	Default or Menu Option	9-6

SECTION 9
IPL - AP101 B

9.1 OVERVIEW

The purpose of the Initial Program Load (IPL) process is to load the GPC with software. The process starts with a GPC that is, for all practical purposes, incapable of doing anything useful. It concludes with a GPC that is fully tested and ready to perform useful work.

The IPL process is divided into two basic phases. The first phase is a hardware phase. This phase is controlled by and implemented in the IOP. The second phase is a software phase. This phase is implemented via software executed by the CPU and IOP.

An overview of the IPL process is provided in figure 9-1.

9.2 HARDWARE PHASE

9.2.1 Introduction

The hardware phase of the IPL process is controlled by the IOP. This portion of the IPL process can be broken down into two distinct phases. The first phase is the hardware response to cockpit switch positions. The second phase is the Microcode execution resulting in the loading of the BOOTSTRAP LOADER.

9.2.2 Hardware Response Phase

The IPL CONTROL LOGIC is IOP hardware that is designed specifically to recognize when an IPL is requested. A functional sketch of this hardware is shown on SSSH drawing 8.3 (zone O6 to N5). The signals it generates, in combination with signals generated by the RESET logic (functional sketch is shown on SSSH drawing 8.3, zone Q7 to O8), initiate and control this phase of the IPL process.

The inputs to IPL CONTROL logic come from the GPC Mode switch, MMU SEL IPL switch, the IPL pushbutton (all located in the cockpit on panel O6), and from the Mass Memory Unit (MMU) (the READY discrete). This logic recognizes an IPL request, determines if the selected MMU is ready, and issues appropriate signals to configure the GPC.

The IPL CONTROL logic recognizes that an IPL is being requested when the GPC mode switch is in HALT, the IPL pushbutton has been depressed, and MMU 1 and/or MMU 2 has been selected for IPL and generates a command to the IOP Microcode called IPL START. The function of this signal will be discussed later when the microcode program operation is detailed.

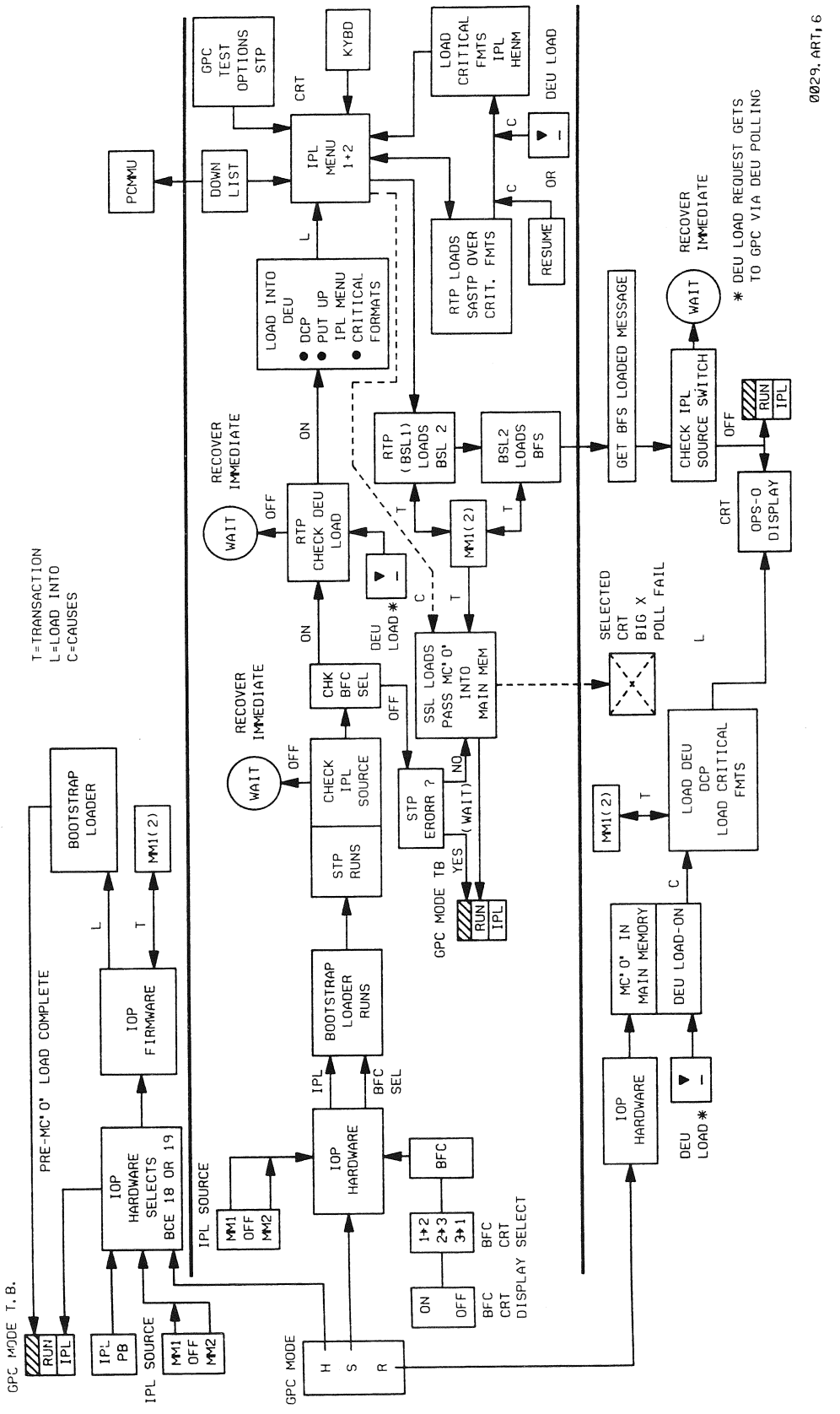


Figure 9-1.- The IPL roadmap.

The IPL CONTROL logic determines that the selected MMU is ready when the READY discrete from the MMU selected by the MMU SEL IPL switch is present. If the selected MMU is ready, the logic generates the signal MM READY. The function of this signal will be discussed later when the microcode program operation is detailed.

The IPL CONTROL logic also configures the GPC for the IPL process. The transmitter and receiver for either BCE 18 (MMU 1 selected for IPL) or BCE 19 (MMU 2 selected for IPL) is enabled. The appropriate BCE is enabled by setting either bit 18 or bit 19 in the HALT/PROCEED register. A signal called GLKOUT(or LOCKOUT) is sent to the CPU and is used to drive the GPC mode talkback to the IPL position.

The SYSTEM RESET logic generates a signal called SYSTEM RESET. While this signal is present, the CPU microcode is stopped. When the IOP hardware configuration is complete, the SYSTEM RESET signal is removed, and the CPU microcode enables the IOP to override the store protect logic (the IOP must load code which is a task it does not normally do) and then enters a loop waiting for the GLKOUT signal to be removed. With the store protect logic suppressed, the IOP must control the state of the store protect bit of each location in memory that it loads.

9.2.3 IOP Microcode Phase

The microcode program that loads the BOOTSTRAP LOADER is specifically designed for that purpose. It is executed by the MSC and either BCE 18 or BCE 19 depending on which BCE has been enabled by the hardware logic.

The microprogram execution begins with the MSC. The MSC first loads a Supervisor Call (SVC) instruction (C9FB) in locations 0-1FFFF. It then determines which BCE (MMU) has been enabled by the hardware logic and sets its BUSY/WAIT bit to BUSY. This allows the selected BCE to begin execution of its microcode program.

The BCE microprogram samples the MM READY signal from the hardware logic. It waits until this signal indicates that the mass memory is ready (not BUSY). When the mass memory is ready, the BCE issues a command to pre-position the tape and then begins loading the software from copy 1 of the BOOTSTRAP LOADER.

When the software has been loaded, it is check-summed. If the checksum is invalid, the BCE will attempt to load copy 2 of the BOOTSTRAP LOADER. If the checksum is once again bad, the microprogram will enter an infinite loop. The only recovery possible is power-cycling the GPC and trying the IPL process again. If the checksum is valid on either the first or second attempt, the IPL process is complete and the BCE will issue the IPLC (IPL COMPLETE) signal. This signal goes to the SYSTEM RESET logic and results in the generation of the SYSTEM RESET signal. This resets all of the registers in the IOP and removes the GLKOUT signal resulting in the mode talkback being undriven (barberpole).

9.3. SOFTWARE PHASE

9.3.1 Introduction

When the hardware-controlled phase of the IPL process is complete, GPC memory has been loaded with the BOOTSTRAP LOADER program. When the GPC mode switch is taken from HALT to STBY, the BOOTSTRAP LOADER program begins execution and loads the GPCIPL program. The GPCIPL program performs a self-test of the GPC and loads either the PASS system software or the Backup System Loader (BSL). The PASS system software is loaded using the Software System Loader (SSL). The selection is based on cockpit switch positions and/or keyboard input.

9.3.2 Bootstrap Loader

After the BOOTSTRAP LOADER has been loaded, the GPC memory has been configured so that, when the GPC Mode switch is taken from HALT to STBY, the BOOTSTRAP LOADER begins execution. Its function is to load the GPCIPL program.

There are three copies of the GPCIPL Program on each MMU. The BOOTSTRAP LOADER will attempt to load copy 1 from the selected MMU. Prior to attempting to load the GPCIPL program, the BOOTSTRAP LOADER will start the Watchdog Timer at its maximum value. Because loading the GPCIPL program takes longer than the maximum value loaded in the Watchdog Timer, the Watchdog Timer will expire. This causes the I-Fail light on the CAM to illuminate, the GPC light on the Caution and Warning Matrix to illuminate, and the Master Alarm to sound. When the load is complete, the BOOTSTRAP LOADER performs a checksum on the GPCIPL software. If the checksum is good, the Watchdog Timer is reset and control is passed to the GPCIPL program.

If the GPCIPL program could not be loaded or the checksum was bad, the BOOTSTRAP LOADER will try to load copy 1 once more. Again, if the checksum is good, control is passed to GPCIPL. Otherwise, the BOOTSTRAP LOADER will attempt to load copy 2 from the selected MMU. It will try twice and, if it is unsuccessful both times, will attempt to load copy 3. Again, it will attempt this twice. If no copy of the GPCIPL program could be loaded, the process will stop with the only recovery available being a power cycle of the GPC and restarting the IPL process.

9.3.3 GPCIPL

9.3.3.1 Introduction

The purpose of the GPCIPL program is to assure that the GPC is healthy by running the Self Test Program (STP) and then to load either the PASS system software or the BSL. The selection of the PASS system software can be done

by default (default option) or by selection from a menu (menu option). The BSL can only be loaded from a menu.

9.3.3.2 Self-Test

The purpose of the STP is to assure that the GPC is healthy and capable of performing as expected. The STP will be run once. It may also be started and stopped from selections on the IPL menu. If it is started from the menu, it will run cyclically until it is stopped. The STP tests 93 percent of the hardware in three functional areas of the GPC: the CPU, the IOP, and memory.

A. CPU tests

The CPU test is done in two sections. The first section uses a CPU instruction to test some of the CPU functions. This instruction allows the testing of the CPU data flow and the real-time counters. The second section uses a software program to test the remainder of the functions which include additional testing of the Arithmetic and Logic Unit (ALU) and the interrupt handling logic.

The CPU data flow tests include testing the internal data paths, general and floating point registers (local store), ALU, microstore, instruction decoding, and addressing capabilities. The data paths between the CPU, IOP, main memory, and the general and floating point registers are tested using alternating bits patterns of ones and zeros. The general and floating point registers are tested to ensure that there are no stuck bits and the registers are properly addressed. The ALU and its associated registers, microcode, and data paths are tested to assure that no bits are stuck and the ALU operations are proper. The macro interrupt and micro interrupt detect logic are tested to assure that they can detect and properly prioritize all interrupts.

The real-time counters are tested to verify that all positions of the counter can be set to either a one or a zero and that an interrupt is generated when the counter decrements from all zeros to all ones.

The software tested functions include additional testing of the ALU and the interrupt handling logic. The ALU is tested using a variety of instruction types to assure that it performs as expected. The interrupt detection logic is tested to assure that all interrupts can be detected and that the proper program status word (PSW) swap occurs.

B. IOP tests

The IOP is tested to assure the proper operation of the channel control, control monitor,and redundancy management logic as well as the proper operation of the MSC/BCE's and their associated registers and data paths. These functional areas are tested using the following general tests. The ability of the CPU to reset the IOP to a known state is tested by issuing a MASTER RESET command. The ability of the CPU to set and reset the Transmitter Enable, Receiver Enable, and Halt/Proceed

registers is tested. The MSC operations are tested using the MSC instruction @STP. This instruction verifies that the MSC can perform all its functions properly. The operation of each BCE is tested using the #STP instruction. All of the BCE functions are tested with the exception of the actual transmission of data by the MIA. The ability of the IOP to generate interrupts and reset them is verified. All IOP-generated interrupts are verified with the exception of the DMA Time Out interrupt and the DMA Queue Overflow interrupt. The Voter Fail Logic and the Watchdog Timer are tested to verify that they can detect failure conditions.

C. Memory tests

Testing of the following functional areas of the GPC memory is provided: core array, read/write control circuitry, timing logic, addressing logic, and the interface to IOP portion of memory. These areas are tested using the following basic tests.

Every memory location is read and checked for parity. The ability to address specific locations is tested. In particular those locations that lie on the upper boundary of each of the 16 pages are tested. The ability to read from and write to both protected and unprotected locations is tested both in proper combinations and improper combinations. Each of the eight-page pairs are tested to assure that they will power down and power up properly.

D. The STP and errors

If the STP encounters errors, its response varies depending on the selection of a default or a menu option load. If the load is a default load, no further action is taken since there is no crew interface. Although, it may be possible to establish a crew interface if the BFC DISPLAY switch is turned ON. If the load is to be done from a menu selection or a crew interface exists, the STP will attempt to display appropriate error messages on the CRT.

Although not an error, the STP cannot complete successfully if a GPC has fail votes against it from other GPC's. Therefore, all fail votes against the GPC being IPLed must be cleared prior to the start of the STP. A list of the error messages that the STP can generate is provided in table 9-I.

9.3.3.3 Default or Menu Option

In order to determine whether the PASS system is to be loaded by default or a menu of software options is to be displayed, the GPCIPL program determines the position of the BFC CRT DISPLAY switch. If the switch is OFF, a default load of PASS software is intended and control is handed to the SSL. If the switch is ON, a menu is to be displayed from which one of three copies of either PASS or BFS software can be selected.

A. Default load

When the STP has completed successfully and the BFC DISPLAY switch is OFF, the loading of PASS Version 1 is started by passing control to the SSL. The SSL will load and check-sum the PASS software. If the checksum is good, the SSL drives the GPC Mode Talkback to the RUN position and monitors the GPC Mode switch for a transition to RUN. When the transition occurs, control is turned over to PASS software. If the checksum is bad, the SSL will attempt to load the software again. If the checksum again fails, it will try one more time. The SSL will attempt to load PASS software a total of three times. If the PASS software cannot be loaded, the only recovery is to restart the IPL process by depressing the IPL pushbutton.

B. Menu option load

If the BFC display switch is ON, a menu option IPL is desired and a crew interface needs to be established. This is accomplished by selecting a DEU using the BFC CRT select switch and activating that DEU's LOAD switch. This will result in the DEU displaying the message DEU LOAD and the DEU being loaded with the software it requires to display the IPL menus. After the DEU has been loaded, the DEU IPL Menu will be displayed. This menu is shown in figure 9-2. There will normally be two messages in the message stack when the IPL MENU first appears. These messages are:

203 - 'GPCIPL XX.XX.XX.XX LOADED'
208 - 'DCP XX.XX.XX.XX.XX LOADED'

These additional messages may appear:

130 - 'NO DEU POLL RESPONSE'
149 - 'DEU BITE ERROR'

Any other messages indicate a problem with the GPC and the IPL should be discontinued. When all of the messages have been acknowledged using the MESSAGE RESET key, one of three copies of either PASS or BFS software may be loaded.

If a copy of PASS software is selected, the SSL will be given control and will attempt, a maximum of three times, to load the selected version of software. When the load is successful, the GPC Mode Talkback will be driven to RUN and the GPC mode switch will be monitored for transition to RUN. When the transition occurs, the PASS software will be given control.

If a copy of the BFS software is selected, the BSL must first be loaded from the mass memory. This is done in two steps. During the first step BSL1 loads the BFS phase table from the MMU. This load is check-summed. If the checksum is good, the title on the IPL MENU changes to BSL1IPL

is also check-summed. If the checksum is good, control is passed to BSL2.

At this point, the title on the IPL menu will change to BSL2IPL MENU. The message BSL 2 AA.BB.CC.DD.EE is driven on the CRT. The AA.BB.CC.DD.EE identifies the version of BSL loaded. When the MSG RESET key is depressed, the BSL will begin loading the selected version of the BFS software. This load is also check-summed. If the checksum is good, the message BFS AA.BB.CC.DD.EE LOADED SET MMU SEL SW TO OFF will be driven on the CRT. The BSL monitors the MMU SEL IPL switch for transition to OFF. When this occurs, the backup flight software is given control and should drive the OPS-000 display.

If any checksum fails, the message MMU COMPUTED CHECKSUM MISCOMPARE will be driven on the CRT. The software begins endlessly computing checksums with the only recovery being a power cycle of the GPC and restarting the IPL process. If the BSL encounters any other error conditions, it will stop and annunciate the error. A list of the errors is provided in table 9-II.

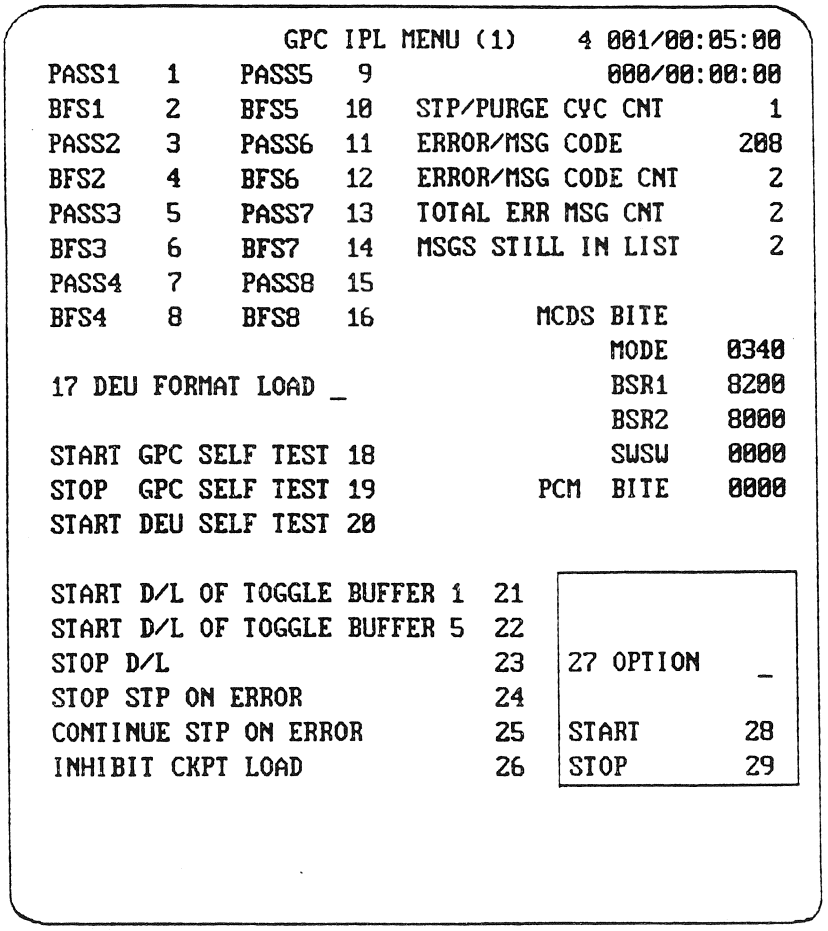


Figure 9-2.- GPC IPL Menu.

TABLE 9-I.- STP ERROR MESSAGES

MSG001-- R/M REGISTER IS NOT ZERO AFTER A MASTER RESET

Explanation: A master reset has failed to reset the bits in the R/M register. These bits must be reset, with the exception of the fail bit and the time out bit.

Contents of Error Information Field (On Page 2 of GPCIPL Menu). R2 will contain the failed bits.

MSG002-- C/M IDLE NOT SET AFTER MASTER RESET

Explanation: The IOP control monitor idle interrupt was not set after a master reset.

Contents of Error Information Field (On Page 2 of GPCIPL Menu). R2 will contain the bits set in interrupt register A.

MSG003-- INT REG A_BITS 1-15 NOT 0 AFTER A MASTER RESET

Explanation: The IOP control monitor idle interrupt was not the only monitor set after a master reset.

Contents of Error Information Field (On Page 2 of GPCIPL Menu). R2 will contain the bits set in interrupt register A. The C/M idle interrupt bit is set to 0.

MSG004-- MASTER RESET DID NOT CLEAR THE TX REGISTER

Explanation: A master reset has failed to reset all MIA transmitters to 0 (disable transmitters).

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which transmitters were not set to 0 (disabled).

MSG005-- MASTER RESET DID NOT CLEAR THE RX REGISTER

Explanation: A master reset has failed to reset all MIA receivers to 0 (disable receivers).

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which receivers were not disabled.

MSG006-- MASTER RESET DID NOT CLEAR THE DO REGISTER

Explanation: A master reset has failed to set discrete output bit 31 (IPL discrete) to 0.

Contents of Error Information Field (On Page 2 off GPCIPL Menu): R2 will show which discretes were not reset to 0.

TABLE 9-I.- Continued

MSG007-- MASTER RESET DID NOT CLEAR THE HALT REGISTER

Explanation: A master reset has failed to disable all processors.

Contents of Error Information Field (On Page 2 off GPCIPL Menu): R2 will show which processors were not disabled.

MSG008— MASTER RESET DID NOT CLEAR THE INTB REGISTER

Explanation: A master reset has failed to reset interrupt register B.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which bits were not reset to 0.

MSG009— MASTER RESET DID NOT CLEAR THE INTC REGISTER

Explanation: A master reset has failed to reset interrupt register C.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which bits were not reset to 0.

MSG0010--MASTER RESET DID NOT CLEAR THE INTD REGISTER

Explanation: A master reset has failed to reset interrupt register D.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which bits were not reset to 0.

MSG0011--MASTER RESET DID NOT CLEAR THE INTE REGISTER

Explanation: A master reset has failed to reset interrupt register E.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which bits were not reset to 0.

MSG0012--MASTER RESET DID NOT CLEAR THE GO/NO-GO REGISTER

Explanation: A master reset has failed to set the GO/NO-GO register to Go (1).

Contents of Error Information Field (On Page 2 of GPCIPL Menu): The state of the GO/NO-GO register is shown in R2. However, the significance of the bits in R2 have been reversed so that all bits set to 1 indicate a NO-GO.

TABLE 9-I.- Continued

MSG013--MASTER RESET DID NOT CLEAR THE BSY/WAIT REGISTER

Explanation: A master reset has failed to set the BUSY/WAIT registers to zero.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which processors are set to one.

MSG014--AN EXPECTED C/M IDLE INTERRUPT DID NOT OCCUR

Explanation: A control/monitor idle interrupt did not occur after a master reset.

MSG015--A PCO_TO RESET BCE HALT BITS_DID NOT RESET ALL

Explanation: The PCO did not enable all processors in KFSTRNG.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the numbers of the processors that were not enabled.

MSG016--A PCO_TO ENABLE BCE XMITTERS_DID NOT ENABLE ALL

Explanation: The PCO did not enable all transmitters in KFSTRNG.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the numbers of the transmitters that were not enabled.

MSG017--A PCO_TO ENABLE BCE RECEIVERS_DID NOT ENABLE ALL

Explanation: The PC did not enable all receivers in KFSTRNG.

Contents of Error Information Field: R2 contains the numbers of the receivers that were not enabled.

MSG018--MSC OR TESTED BCE IS NO-GO

Explanation: After the IOP self test module is processed, the MSC on at least one of the tested BCE's has remained set to NO-GO.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): If the terminate switch is OFF, R2 must be set to FFFF $FF80_{16}$. If the terminate switch is ON, R2 must be set to FFFC 3080_{16}.

MSG019--MSC OR TESTED BCE STILL BUSY

Explanation: One or more of the processors tested did not go to the WAIT state.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 will show which processors are not in the WAIT state.

TABLE 9-I.- Continued

MSG020--AN EXPECTED EXTERNAL INTERRUPT 2 DID NOT OCCUR

Explanation: The @INT instructions did not cause an external 2 interrupt.

MSG021--MSC COULD NOT TALLY HIGHEST FW MEMORY LOCATION

Explanation: The MSC executed an unexpected @WAIT instruction for one of two reasons:

1. Location $33FFE_{16}$ did not contain $FFFFFFFF_{16}$.
2. The @TSZ instruction failed.

MSG022--AN EXPECTED STORE-PROTECT INTRRPT DID NOT OCCUR

Explanation: Attempt to store data into a protected location failed to cause program interrupt.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the address of the failed location.

MSG023--STORE PROTECT INTRRPT ON AN UNPROTECTED LOCATION

Explanation: The storage of data into an unprotected location caused a program interrupt.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the address of the unprotected location.

MSG024--MEMORY ADDRESSING ERROR

Explanation: One of the bits in the storage address register did not change from 1 to 0. The address is generated by combining R2 and R7 through an "EXCLUSIVE OR".

MSG025--SYSTEM SOFTWARE LOADER CHECKSUM ERROR

Explanation: The interrupt was the result of a bad software system loader checksum. The computed SSL checksum is in R4.

MSG026--MASTER RESET DID NOT CAUSE AN INTERRUPT

Explanation: Master reset must cause a control/monitor idle interrupt.

MSG027--MCH CHK: EXTENDED STORAGE ADDRESS PARITY

Explanation: A parity error was found in extended memory, locations 28000_{16} to $33FFF_{16}$.

TABLE 9-I.- Continued

MSG028--MCH CHK: IOP STORAGE PARITY

Explanation: IOP (encountered) storage parity for main store access in either the IOP or CPU units.

MSG029--MCH CHK: CPU STORAGE PARITY

Explanation: CPU (encountered) storage parity for main store access in the CPU unit.

MSG030--MCH CHK: EXTENDED STORAGE DATA PARITY

Explanation: A data parity error was found in extended memory, locations 28000_{16} to $33FFF_{16}$.

MSG031--MCH CHK: CPU ROS PARITY

Explanation: A CPU ROS parity error was found.

MSG032--MCH CHK: INTERRUPT CODE NOT IN THE RANGE 1-5

Explanation: The machine check interrupt code was greater than 5 or less than 1.

MSG033--NOT ASSIGNED---USER MAY FILL IN A MESSAGE

MSG034--PGM CHK: PRIVILEGED INSTRUCTION

Explanation: A privileged instruction was executed while in the problem state.

MSG035--PGM CHK: INTERRUPT CODE 2 IS NOT DEFINED

Explanation: A program interrupt was taken with an illegal interrupt code.

MSG036--PGM CHECK: CPU ADDRESS SPECIFICATION ERROR

Explanation: The main store operation performed by the CPU used a non-existent storage location. This interrupt is not maskable.

MSG037--PGM CHECK: FIXED POINT OVERFLOW

Explanation: The result of a fixed point arithmetic operation is too large and will not fit into a full word format.

MSG038--PGM CHECK: SIGNIFICANCE ERROR

Explanation: The fraction that resulted from a floating point operation is zero, while the characteristic is not zero.

TABLE 9-I.- Continued

MSG039--PGM CHECK: UN-NORMALIZED INPUTS-FLTNG POINT DVD -

Explanation: The input data to a floating point divide is not normalized. There is no mask for this interrupt.

MSG040--PGM CHK: CPU STORE PROTECT VIOLATION

Explanation: A memory write operation was attempted into one of the main memory locations that have an active store protect bit.

MSG041--PGM CHK: INTERRUPT CODE 8 IS NOT DEFINED

Explanation: A program interrupt was taken with an illegal interrupt code.

MSG042--PGM CHK: EXPONENT UNDERFLOW-FLOATING POINT

Explanation: The characteristic that resulted from a floating point arithmetic operation was negative and could not be expressed in the 7-bit format allocated to the characteristic.

MSG043--PGM CHK: EXPONENT OVERFLOW-CONVERT INSTRUCTION

Explanation: The characteristic that resulted from a CVFL instruction is greater than 127 and the fraction is not 0. There is no mask for this interrupt.

MSG044-PGM CHECK: EXPONENT OVERFLOW-FLOATING POINT-

Explanation: The characteristic that resulted from a floating point arithmetic operation is greater than 127 and the fraction is not 0.

MSG045--PGM CHK: DIVIDE INSTRUCTION-FLOATING-

Explanation: The denominator used by a floating point divide instruction is equal to 0.

MSG046--PGM CHK: INTERRUPT CODE GREATER THAN 12

Explanation: A program interrupt was taken with an illegal interrupt code.

MSG047--UNEXPECTED SVC INTERRUPT

Explanation: An unexpected SVC interrupt was taken with an invalid interrupt code.

MSG048--UNEXPECTED CLOCK 1 INTERRUPT

Explanation: The low 16-bit hardware binary counter causes a macro interrupt when the interrupt is not masked and the counter is decremented to zero (maximum of 65.536ms). This causes a PSW swap only if the high counter (location $00B0_{16}$) is also zero.

TABLE 9-I.- Continued

MSG049--UNEXPECTED CLOCK 2 INTERRUPT

Explanation: The low 16-bit hardware binary counter causes a macro interrupt when the interrupt is not masked and the counter is decremented down to 0 (maximum of 65.536ms). This causes a PSW swap only if the high counter (location $00B1_{16}$) is also 0.

MSG050--UNEXPECTED INSTRUCTION MONITOR INTERRUPT

Explanation: The instruction that was fetched, from the main store during a macro instruction operation, is not store protected.

MSG051--UNEXPECTED EXTERNAL 0 INTERRUPT

Explanation: The interrupt register A was not set when an external 0 interrupt occurred.

MG052--UNEXPECTED EXTRNL 1 INTRPT: INTRPT CODE = 0

Explanation: The interrupt was requested by the IOP via the CPU/IOP interface. The interrupt register B must be read to find what caused the error. The register may be found in the downlist.

MSG053--UNEXPECTED EXTRNL 1 INTRPT: PCl DATA PARITY

Explanation: The data received by the CPU at the CPU/IOP interface, through a program controlled in (PCI) instruction, contains a parity error.

MSG054--UNEXPECTED EXTRNL 1 INTRPT: I/O WRITE PARITY

Explanation: The data received by the CPU at the CPU/IOP interface, during a direct memory access (DMA) operation, contains a parity error.

MSG055--UNEXPECTED EXTRNL 1 INTRPT: I/O ADDRESS SPEC ERROR

Explanation: The I/O used a non existent (or out of bounds) main storage address during a DMA operation.

MSG056--UNEXPECTED EXTRNL 1 INTRPT: I/O STORE-PROTECT

Explanation: This interrupt occurs if a DMA IN operation causes the main store to write into a protected location.

MSG057--UNEXPECTED EXTRNL 1 INTRPT: I/O ADDRESS PARITY

Explanation: The address received by the CPU at the CPU/IOP interface, during a DMA operation, contains a parity error.

MSG058--UNEXPECTED EXTRNL 1 INTRPT: AGE INTERRUPT

Explanation: The AGE line at the CPU/AGE interface was activated.

TABLE 9-I.- Continued

MSG059--UNEXPECTED EXTERNAL 2 INTERRUPT

Explanation: The interrupt was caused by the @INT instruction.

MSG060--UNEXPECTED EXTERNAL 3 INTERRUPT

Explanation: This interrupt is not used except during testing.

MSG061--UNEXPECTED EXTERNAL 4 INTERRUPT

Explanation: This interrupt is not used except during testing.

MSG062--BCE TIME-OUT N RETRY FAIL-DEUIPL

Explanation: Interrupt register C is set to 2000 0000_{16}, indicating a BCE timeout has taken place during a DEU IPL.

MSG063--GPC ID VALUE NOT IN RANGE 1-5

Explanation: Interrupt register C is set to 2000 0000_{16}, indicating a BCE interrupt has occurred during a DEU IPL.

MG064--CONDITION CODE NOT ZERO AFTER A FULL DETECT TEST

Explanation: The test performs the regular DETECT option of the CPU DETECT instruction. The condition code in the old PSW must be checked for the following errors:

01-- BITE error (ROS parity, micro and macro interrupts)

11-- Data flow and ALU error.

MSG065--CONDITION CODE NOT ZERO AFTER A COUNTERS TEST

Explanation: Condition code of 11 signifies that counter 1 or counter 2 failed.

MSG066--REG 4 NOT LOADED CORRECTLY VIA R2-TO-R3-TO-R4

Explanation: R4 does not equal R3. The following condition must hold: R2=R3=R4=R5=R6=1111 0000_{16}

MSG067--THE SUM FORMED IN R2 IS TOO HIGH

Explanation: The contents of one or more of the registers listed below is not correct. The contents should be as follows:

R2 = CCCC 0000_{16} R4 = 3333 0000_{16} R6 = 1111 0000_{16}
R3 = 4444 0000_{16} R5 = 2222 0000_{16}

TABLE 9-I.- Continued

MSG068--THE SUM FORMED IN R2 IS TOO LOW

Explanation: The contents of one or more of the registers listed below is not correct. The contents should be as follows:

R2 = CCCC 0000_{16} R4 = 3333 0000_{16} R6 = 1111 0000_{16}
R3 = 4444 0000_{16} R5 = 2222 0000_{16}

MSG069--THE SHIFT TEST FAILED

Explanation: R2 is not equal to R5. Both R2 and R5 must contain 2222 0000_{16}

MSG070--BHE/BL INSTRUCTION OR CONDITION CODE FAILURE

Explanation: The test failed because the contents of R2 was greater than or equal to the contents of R4. The registers must contain the following:

R2 = 2222 0000_{16}

R4 = 3333 0000_{16}

The condition code is not 11.

MSG071--BLE INSTRUCTION FAILED TO BRANCH

Explanation: The test failed because the contents of R2 was greater than the contents of R4. The registers must contain the following:

R2 = 2222 0000_{16}

R4 = 3333 0000_{16}

The condition code is not 11.

MSG072--BLE/BH INSTRUCTION OR CONDITION CODE FAILURE

Explanation: The test failed because the contents of R4 was less than or equal to the contents of R2. The contents should be as follows:

R2 = 2222 0000_{16}

R4 = 4444 0000_{16}

The condition code was not set to 01.

TABLE 9-I.- Continued

MSG073--BHE INSTRUCTION OR CONDITION CODE FAILURE

Explanation: The test failed because the contents of R4 was less than the contents of R2. The contents should be as follows:

R2 = 2222 0000_{16}

R4 = 4444 0000_{16}

The condition code was not set to 01.

MSG074--BM INSTRUCTION OR CONDITION CODE FAILURE

Explanation: The condition code was not set to 11.

MSG075--BE INSTRUCTION OR CONDITION CODE FAILURE

Explanation: R3 is not set equal to 7F33 0000_{16}. The condition code is not set to 00.

MSG076--TRB/BO INSTRUCTION OR CONDITION CODE FAILURE

Explanation: R3 is not set equal to 7F33 0000_{16}. The condition code is not set to 01.

MSG077--OHI INSTRUCTION FAILURE

Explanation: R3 is not set equal to FFFF 0000_{16}. The condition code is not set to 00.

MSG078--OVERFLOW NOT SET OR BOV INSTRUCTION FAILURE

Explanation: R3 is not set equal to 0001 0000_{16} The overflow indicator is not set.

MSG079--CARRY BIT NOT SET OR BOC INSTRUCTION FAILURE

Explanation: R3 is not equal to 0001 0000_{16}. The carry indicator is not set.

MSG080--SIGN BIT NOT SET OR BP INSTRUCTION FAILURE

Explanation: R3 is not greater than 0. The register is not set to 0001 0000_{16}. The condition code is not equal to 01.

MSG081--LACR INSTRCTN DID NOT SET CC TO INDICATE LT 0

Explanation: R3 is not less than 0. The register is not set to FFFF 0000_{16}. The condition code is not equal to 11.

TABLE 9-I.- Continued

MSG082--AH/BP/BZ INSTRUCTION OR CONDITION CODE FAILURE

Explanation: R3 is not set to 0. The condition code is not set to 00.

MSG083--TS --TEST AND SET-- INSTRUCTION FAILURE

Explanation: The TS@ # instruction did not set the condition code to 00.

MSG084--TB/BO INSTRUCTION OR CONDITION CODE FAILURE

Explanation: R2 is not equal to FFFF 0000_{16}. The condition code is not set to 01.

MSG085--BVC INSTRUCTION OR CONDITION CODE FAILURE

Explanation: The carry indicator is not set. R4 is C3FF $421F_{16}$.

MSG086--DR/MHI/CHI INSTRUCTION OR COND CODE FAILURE

Explanation: R4 is not equal to 1649 0000_{16}. The condition code is not set to 00.

MSG087--FLOATING POINT TEST 1 FAILURE

Explanation: F2 is not equal to F7. Both registers should read 0000 0000_{16}.

MSG088--FLOATING POINT TEST 2 FAILURE

Explanation: The contents of F4 or F5 are not correct.

F4 must contain 4412 3431_{16}.

F5 must contaiN BA66 6541_{16}.

MSG089--FLOATING POINT TEST 3 FAILURE

Explanation: F3 is not equal to EF1D $DDDD_{16}$.

MSG090--FLOATING POINT TEST 4 FAILURE

Explanation: F2 is not equal to 0.

MSG091--FLOATING POINT TEST 5 FAILURE

Explanation: F2 and F3 are not set to 0.

TABLE 9-I.- Continued

MSG092--FLOATING POINT TEST 6 FAILURE

Explanation: The contents of F0 or F1 were not correct. F0 and F1 should respectively contain 4146 $2D5A_{16}$ and 2910 5916_{16}.

MSG093--BRANCH INDIRECT FAILED

Explanation: The branch indirect instruction failed.

MSG094--SRDR SHIFT FAILED

Explanation: The SRDR instruction failed. R7 must be equal to 0000 6000_{16}.

MSG095--ADDRESSING INTRPT FAILURE ON ILLEGAL ADDRESS

Explanation: An access to an address outside of core did not cause a program check interrupt.

MSG096--STORE-PROTECT INTERRUPT FAILED TO OCCUR

Explanation: Attempt to store into a protected location did not cause a program interrupt.

MSG097--STORE-PROTECT INTERRUPT CODE NOT 7

Explanation: The program interrupt code was not set equal to 7 by the store protect (SP) interrupt.

MSG098--INT CODE NOT HEX F AFTER INS MON INTERRUPT

Explanation: The interrupt code set by the instruction monitor interrupt handler is not equal to 15.

MSG099--FIXED POINT OVERFLOW NOT MASKABLE

Explanation: This interrupt is caused by a fixed point overflow with program interrupts masked.

MSG100--FIXED POINT OVERFLOW INTERRUPT FAILED TO OCCUR

Explanation: Fixed point overflow with program interrupts not masked failed to cause an interrupt.

MSG101--INT CODE NOT 4 AFTER FIXED PT OVFLO INTERRUPT

Explanation: The interrupt code after an expected fixed point overflow program interrupt, is not equal to 4.

TABLE 9-I.- Continued

MSG102--MCH CHK INTRPT NOT MASKABLE WITH CPU MEM PARITY

Explanation: The reading of a bad parity word with the machine check interrupt masked caused an unexpected CPU memory parity error interrupt.

MSG103--MCH CHK INTRPT DID NOT OCCUR ON CPU MEM PARITY

Explanation: The reading of a bad data parity word with the machine check interrupt unmasked failed to cause the required machine check interrupt.

MSG104--MCH CHK INT CODE NOT 3 AFTER CPU MEM PARITY

Explanation: The interrupt code after an expected CPU memory parity check is not 3.

MSG105--R/M VOTER TEST 1 FAILURE

Explanation: One of the test failure vote inputs failed. Alternately, the test fail vote that inhibits program controlled outputs (PCO) has failed. Expected content of RM Status Register is 401E 0000 after being masked by E01F E000.

MSG106--COULD NOT SET TIME OUT LATCH WITH WATCHDOG

Explanation: A forced time out of the watchdog timer did not set the fail latch in the RM status register.

MSG107--DID NOT GET AN EXPECTED EXT 0 INTERRUPT

Explanation: A forced time out of the watchdog timer did not cause an external 0 interrupt.

MSG108--WATCHDOG INT BIT NOT ON AFTER EXT 0 INTERRUPT

Explanation: A forced time out of the Watchdog Timer failed to set the Watchdog Timer bit (bit 0) of the Interrupt Register A. The status of Interrupt Register A is stored in R4.

MSG109--EXPECTED EXT 4 INTERRUPT DID NOT OCCUR

Explanation: An expected external 4 interrupt did not occur after a test interrupt PCO instruction.

MSG110--EXPECTED EXT 3 INTERRUPT DID NOT OCCUR

Explanation: An expected external 3 interrupt did not occur after a test interrupt PCO instruction.

MSG111--EXPECTED EXT 1 INTERRUPT DID NOT OCCUR

Explanation: An expected external 1 interrupt did not occur after a test interrupt PCO instruction.

TABLE 9-I.- Continued

MSG112--EXPECTED EXT 0 INTERRUPT DID NOT OCCUR

Explanation: An expected external 0 interrupt did not occur after a test interrupt PCO instruction.

MSG113--UNEXPECTED EXTRN 1 INTRPT-INTERRUPT CODE > 6

Explanation: An illegal interrupt code caused an external 1 interrupt.

MSG114--UNEXPECTED INT REG E VALUE AFTER EXT 4 INTERRUPT

Explanation: Interrupt register E is not reset. All interrupts must be reset after a test interrupt PCO instruction.

MSG115--UNEXPECTED INT REG D VALUE AFTER EXT 3 INTERRUPT

Explanation: Interrupt register D is not reset. All interrupts must be reset after a test interrupt PCO instruction.

MSG116--UNEXPECTED INT REG B VALUE AFTER EXT 1 INTERRUPT

Explanation: Interrupt register B is not reset. All interrupts must be reset after a test interrupt PCO instruction.

MSG117--UNEXPECTED INT REG A VALUE AFTER EXT 0 INTERRUPT

Explanation: Interrupt register A is not reset. All interrupts must be reset after a test interrupt PCO instruction.

MSG118--NO EXPECTD EXT 1 INT AFTER BAD PRTY WRITE BY MSC

Explanation: The entry of data with bad parity into CPU core by the master sequence controller (MSC) failed to cause an external 1 interrupt. The PCO instruction with bad parity data input enabled failed to cause a bad parity interrupt.

MSG119--IOP REG NOT 0 AFTER EXT 1 BAD PARITY WRITE INT

Explanation: The interrupt register B must not be set after an external 1 interrupt was triggered by a bad parity write.

MSG120--INT CODE NOT 1 AFTER EXT 1 PCI DATA PARITY INT

Explanation: The interrupt code was not set to 1 after a data parity error was found during the execution of a PCI instruction.

MSG121--NO EXPECTED EXT 1 INT ON BAD DMA ADDRESS PARITY

Explanation: A direct memory access (DMA) address parity error failed to cause an external 1 interrupt. The PCO instruction with bad DMA address parity failed.

TABLE 9-I.- Continued

MSG122--INT CODE NOT 5 AFTER EXT 1 DMA ADDRS PARITY INT

Explanation: Interrupt code must be set to 5 after an external 1 interrupt is triggered by a bad direct memory access (DMA) address parity.

MSG123--IOP REG NOT 0 AFTER EXT 1 DMA ADDRS PARITY INT

Explanation: Interrupt register B must not be set after an external 1 interrupt is triggered by a bad DMA address parity.

MSG124--GPC POWER FAIL AT LOCATION SHOWN - RECOVERED OK

Explanation: A power failure occurred. The power down PSW is in the error environment stack.

MSG125--MSC @STP OR BCE #STP WRAP FAILED

Explanation: One or more of the processors tested failed the read/write part of the @STP or the #STP instruction.

MSG126--MSC NOT IN WAIT AFTER BAD ADDRESS PARITY TEST

Explanation: MSC status register 1 must be set to no-go after a bad address parity.

MSG127--NO EXPECTED MCH CHK IOP STORAGE PARITY INTERRUPT

Explanation: Reading from locations 2,3 failed to cause a machine check interrupt. The location contains data with a parity error.

MSG128--INT CODE NOT 2 AFTER MCH CHK IOP STRGE PRITY INT

Explanation: Machine check interrupt code must be 2 after an IOP storage data parity interupt.

MSG129--REAL TIME MSC TIME OUT

Explanation: The contents of interrupt register C is equal to $1000\ 0000_{16}$. This indicates an MSC time out during DEU IPL.

MSG130-NO DEU POLL RESPONSE

Explanation: The DEU did not send a poll response message after it was polled by the GPC.

MSG131--REAL TIME IPL RETRY FAILURE

Explanation: The contents of the interrupt register C is equal to $3000\ 0000_{16}$. This indicates that a page of the DCP did not successfully transfer from the GPC to the DEU.

TABLE 9-I.- Continued

MSG132--DEU INITIALIZATION FAILURE

Explanation: Interrupt register C is set to 4000 0000_{16}. This setting indicates that the DEU status registers reflected a bad load of the DCP.

MSG133--UNEXPECTED EXT 0 INTRPT: CODE = 0 /WATCHDOG TIMER/

Explanation: The GO/NO-GO timer (watchdog timer) has timed out and generated an external 0 interrupt.

MSG134--UNEXPECTED EXT 0 INTRPT: CODE = 0 /IOP FAIL LATCH/

Explanation: The redundancy management (RM) voter logic has detected a failure that affects the capablility of the machine.

MSG135--UNEXPCTD EXT 0 INTRP: CODE = 0 CNTRL/MONITR IDLE

Explanation: An unexpected control monitor idle interrupt has occurred.

MSG136--UNEXPECTED EXT 0 INTRP: CODE = 0 IOP ROS PRITY ERROR

Explanation: A data parity error has occurred during transfer from IOP read only storage (ROS).

MSG137--UNEXPECTED EXT 0 INTRP: CODE = 0 /IOP FAULT/

Explanation: The input output oscillator is stopped.

MSG138--R/M VOTER TEST 2 FAILURE

Explanation: One or more of the inputs to the load test register PCO instruction did not set the fail or time out bit in the RM status register as expected (expected contents of register after being masked by E01F E000 is C001 0000_{16}). The inputs are as follows:

Voter Test Inputs 1 and 2 = 0000 $001C_{16}$

Voter Test Inputs 1 and 3 = 0000 $001A_{16}$

Voter Test Inputs 1 and 4 = 0000 0019_{16}

Voter Test Inputs 2 and 3 = 0000 0016_{16}

Voter Test Inputs 2 and 4 = 0000 0015_{16}

Voter Test Inputs 3 and 4 = 0000 0013_{16}

Bit 27 inhibits the normal voter inputs from the other input output processors, and inhibits the driving of the computer fail latch and the IOP transmission termination logic.

TABLE 9-I.- Continued

MSG139--I/O STORE PROTECT TEST FAILED

Explanation: The DMA Store Protect Interrupt Test failed. The test consists of a write attempt by the IOP to a store-protected location in the CPU.

The contents of the following registers indicate the nature of the error:

Register	Explanation
R5 bit 15=1	Expected external 1 interrupt did not occur.
R5 bit 14=1	MSC was not in the WAIT state.
R5 bit 13=1	MSC program counter incorrect.
R5 bit 12=1	External 1 interrupt return code is not equal to 4.
R3 bits 0-15	External 1 interrupt return code.
R3 bits 16-31	Interrupt register B.
R6 bits 14-31	MSC program counter, 18-bits.

MSG140--COULD NOT RESET ALL TERMINATE CONTROL LATCHES

Explanation: After the termination control latches are set and reset by the PC instruction, the redundancy management (RM) status register must read $0000\ 6000_{16}$. This setting means that both the voter termination control latch and the timer termination control latch are set. R5 must contain the same setting as does the RM status register.

MSG141--UNEXPECTED IOP PROGRAMMED INTERRUPT

Explanation: An external 2 interrupt occurred during a DEU IPL. Interrupt Register C is set to an unexpected value. The contents of R3 must equal the content of interrupt register C.

MSG142--IOP ROS PARITY TEST FAILED

Explanation: The contents of the following registers indicate the nature of the error:

Register	Explanation
R5 bit 15=1	Expected external 0 interrupt did not occur.
R5 bit 14=1	MSC was not in the HALT state.
R5 bit 13=1	MSC program counter incorrect.
R5 bit 12=1	Interrupt register A bit 3, IOP ROS parity, was not set.
R3 bits 0-15	External 0 interrupt return code.
R3 bits 16-31	Interrupt register A.
R6 bits 14-31	MSC program counter, 18-bits.

MSG143--THE INSTRUCTION MONITOR IS NOT MASKABLE

Explanation: An interrupt was caused by the excecution of an unprotected instruction with the the instruction monitor interrupt mask set to off. The mask is set to off if bit 34 of the PSW is set to 0.

TABLE 9-I.- Continued

MSG144--THE INS MON DID NOT INTRPT ON AN UNPROT INSTRCTN

Explanation: No interrupt was caused by the execution of an unprotected instruction while the instruction monitor mask is set to on. The mask is on if bit 34 of the PSW is set to 1.

MSG145--IOP RET NOT 0 AFTER EXT 1 PCI DATA PARITY INT

Explanation: Interrupt register B must be set to 0 after a data parity error is found during the execution of a PCI instruction. The contents of R4 must equal the contents of interrupt register B.

MSG146--NO EXPECTED EXT 1 INT AFTER PCI DATA PARITY

Explanation: A parity error in the PCI data failed to cause an external 1 interrupt. The bad parity data input PCO instruction failed to force a parity error on data transferred to the CPU.

MSG147--INT CODE NOT 2 AFTER EXT 1 PCI DATA PARITY INT

Explanation: A DMA data write containing a parity error failed to set external 1 interrupt condition code to 2.

MSG148--MEMORY PARITY TEST - PARITY ERROR

Explanation: An unexpected machine check occurred during the memory parity test.

MSG149--DEU BITE ERROR

Explanation: The BITE error bit was set in the header word of the DEU poll response. The contents of the following registers contain the DEU response data for the DEU BITE:

Register	Explanation
R4 bits 0-15	DEU Header Word (see Appendix III)
R5 bits 0-15	DEU Software Status Word (see Apendix III)
R6 bits 0-15	DEU BITE Status Word 1 (see Appendix III)
R6 bits 16-31	DEU BITE Status Word 2 (see Apendix III)

MSG150--STARTUP SELF - TEST ABENDED WITH TOO MANY ERRORS

Explanation: The Self Test Program, which is run at startup, resulted in more that 98 errors in a single run.

MSG151--MMU COMPUTED CHECK SUM MISCOMPARE

Explanation: The checksum furnished by the MMU was not equal to the computed checksum. The computed checksum is stored in R5.

TABLE 9-I.- Continued

MSG152--MMU WILL NOT GO READY

Explanation: The MMU ready discrete bit was not set in discrete input A (DIA). Bit 6 of DIA indicates that MMU 1 is available for use and bit 7 indicates that MMU 2 is available for use.

MSG153--MMU EXPECTED POSITION ERROR

Explanation: After each position or read command sent to the MMU, the position of the MMU is interrogated through a position request command. The warning message results when the position given in the MMU response does not correspond to the expected position.

The contents of the following registers indicate the nature of the error:

Register	Explanation
R3 bits 2-12	Actual MMU position.
R6 bits 2-12	Computed MMU position.

MSG154--MMU SOURCE SELECT SWITCH POSITION ERROR

Explanation: The IPL source select switch indicates both MMU's are selected or no MMU is selected. Only one MMU should be selected.

The contents of the following registers indicate the nature of the error:

Register	Explanation
R5 bit 4=1	MMU 1 selected.
R5 bit 5=1	MMU 2 selected.

MSG155--MMU BITE STATUS REG A OR B ERROR

Explanation: The MMU status registers were not set to zero after a MMU status command. R3 contains the MMU status.

MSG156--MMU BCE I/O ERROR

Explanation: The No-Go status was set for BCE 18 or BCE 19, where BCE 18 corresponds with MMU 1 and BCE 19 to MMU 2.

MSG157--MSC STATUS ERROR AFTER BAD ADDRESS PARITY TEST

Explanation: The test failed to set either bits 15 or 16 of the MSC status after forcing a Bad Address Parity. Bit 16 indicates that the execution of a previous command resulted in an error. Bit 15 indicates the use of an illegal op-code.

Contents of Error Information Field (On Page 2 of the GPCIPL Menu):
R7 contains the MSC status.

MSG158--MSC STAT 1 ERROR AFTER BAD ADDRESS PARITY TEST

Explanation: The reset status 1 (Go/No-Go) PCO operation followed by the read status PCI operation failed to set the MSC to go.

TABLE 9-I.- Continued

MSG159--INSTRUCTION MONITOR INTERRUPT ON PROTECTED INST.

Explanation: The processing of a protected instruction caused an instruction monitor interrupt.

MSG160--POWER ON SWITCHING TEST FAILURE PAGE 1 C A16

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A15/C-A16, (sector 0, addresses = 0-7FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG161--POWER ON SWITCHING TEST FAILURE PAGE 2 C A15

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A15/C-A16, (sector 0, addresses = 0-7FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG162--POWER ON SWITCHING TEST FAILURE PAGE 3 C A18

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A17/C-A18, (sector 1, addresses = 8000-FFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG163--POWER ON SWITCHING TEST FAILURE PAGE 4 C A17

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A17/C-A18, (sector 1, addresses = 8000-FFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG164--POWER ON SWITCHING TEST FAILURE PAGE 5 C A20

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A19/C-A20, (sector 2, addresses = 10000-17FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

TABLE 9-I.- Continued

MSG165--POWER ON SWITCHING TEST FAILURE PAGE 6 C A19

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A19/C-A20, (sector 2, addresses = 10000-17FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG166--POWER ON SWITCHING TEST FAILURE PAGE 7 C A22

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A21/C-A22, (sector 3, addresses = 18000-1FFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG167--POWER ON SWITCHING TEST FAILURE PAGE 8 C A21

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A21/C-A22, (sector 3, addresses = 18000-1FFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG168--POWER ON SWITCHING TEST FAILURE PAGE 9 C A24

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A23/C-A24, (sector 4, addresses = 20000-27FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG169--POWER ON SWITCHING TEST FAILURE PAGE 10 C A23

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the CPU memory page pair C-A23/C-A24, (sector 4, addresses = 20000-27FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG170--POWER ON SWITCHING TEST FAILURE PAGE 11 I A27

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the IOP memory page pair I-A26/I-A27, (sector 5 lower, addresses = 28000-2BFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

TABLE 9-I.- Continued

MSG171--POWER ON SWITCHING TEST FAILURE PAGE 12 I A26

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the IOP memory page pair I-A26/I-A27, (sector 5 lower, addresses = 28000-2BFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG172--POWER ON SWITCHING TEST FAILURE PAGE 13 I A25

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the IOP memory page pair I-A24/I-A25, (sector 5 upper, addresses = 2C000-2FFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG173--POWER ON SWITCHING TEST FAILURE PAGE 14 I A24

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the IOP memory page pair I-A24/I-A25, (sector 5 upper, addresses = 2C000-2FFFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG174--POWER ON SWITCHING TEST FAILURE PAGE 15 I A23

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the IOP memory page pair I-A22/I-A23, (sector 6 lower, addresses = 30000-33FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG175--POWER ON SWITCHING TEST FAILURE PAGE 16 I A22

Explanation: The average cycle time of the power switching test routine exceeded 12.9 μsec, over 50 passes. This indicates that the IOP memory page pair I-A22/I-A23, (sector 6 lower, addresses = 30000-33FFF), failed to remain in Standby for the complete cycle duration as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG176--POWER OFF SWITCHING TEST FAILURE PAGE 1 C A16

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A15/C-A16, (sector 0, addresses = 0-7FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

TABLE 9-I.- Continued

MSG177--POWER OFF SWITCHING TEST FAILURE PAGE 2 C A15

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A15/C-A16, (sector 0, addresses = 0-7FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG178--POWER OFF SWITCHING TEST FAILURE PAGE 3 C A18

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A17/C-A18, (sector 1, addresses = 8000-FFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG179--POWER OFF SWITCHING TEST FAILURE PAGE 4 C A17

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A17/C-A18, (sector 1, addresses = 8000-FFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG180--POWER OFF SWITCHING TEST FAILURE PAGE 5 C A20

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A19/C-A20, (sector 2, addresses = 10000-17FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG181--POWER OFF SWITCHING TEST FAILURE PAGE 6 C A19

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A19/C-A20, (sector 2, addresses = 10000-17FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

TABLE 9-I.- Continued

MSG182--POWER OFF SWITCHING TEST FAILURE PAGE 7 C A22

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A21/C-A22, (sector 3, addresses = 18000-1FFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG183--POWER OFF SWITCHING TEST FAILURE PAGE 8 C A21

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A21/C-A22, (sector 3, addresses = 18000-1FFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG184--POWER OFF SWITCHING TEST FAILURE PAGE 9 C A24

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A23/C-A24, (sector 4, addresses = 20000-27FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG185--POWER OFF SWITCHING TEST FAILURE PAGE 10 C A23

Explanation: The average cycle time of the power switching test routine is less than 25.4 μsec, over 50 passes. This indicates that the CPU memory page pair C-A23/C-A24, (sector 4, addresses = 20000-27FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the CPU timing page maybe malfunctioning.

MSG186--POWER OFF SWITCHING TEST FAILURE PAGE 11 I A27

Explanation: The average cycle time of the power switching test routine is less than 25.5 μsec, over 50 passes. This indicates that the IOP memory page pair I-A26/I-A27, (sector 5 lower, addresses = 28000-2BFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

TABLE 9-I.- Continued

MSG187--POWER OFF SWITCHING TEST FAILURE PAGE 12 I A26

Explanation: The average cycle time of the power switching test routine is less than 25.5 μsec, over 50 passes. This indicates that the IOP memory page pair I-A26/I-A27, (sector 5 lower, addresses = 28000-2BFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG188--POWER OFF SWITCHING TEST FAILURE PAGE 13 I A25

Explanation: The average cycle time of the power switching test routine is less than 25.5 μsec, over 50 passes. This indicates that the IOP memory page pair I-A24/I-A25, (sector 5 upper, addresses = 2C000-2FFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG189--POWER OFF SWITCHING TEST FAILURE PAGE 14 I A24

Explanation: The average cycle time of the power switching test routine is less than 25.5 μsec, over 50 passes. This indicates that the IOP memory page pair I-A24/I-A25, (sector 5 upper, addresses = 2C000-2FFFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG190--POWER OFF SWITCHING TEST FAILURE PAGE 15 I A23

Explanation: The average cycle time of the power switching test routine is less than 25.5 μsec, over 50 passes. This indicates that the IOP memory page pair I-A22/I-A23, (sector 6 lower, addresses = 30000-33FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

MSG191--POWER OFF SWITCHING TEST FAILURE PAGE 16 I A22

Explanation: The average cycle time of the power switching test routine is less than 25.5 μsec, over 50 passes. This indicates that the IOP memory page pair I-22/I-A23, (sector 6 lower, addresses = 3000-33FFF), remained in Standby for the cycle duration rather than powering down to Quiescent state as expected. The corresponding one shot in the IOP timing page maybe malfunctioning.

TABLE 9-I.- Continued

MSG192--BCE WRAP-#STP ERR-AT LEAST 1 BCE DID NOT FINISH

Explanation: One or more of the processors tested failed to go to wait.

MSG193--MSC-@STP WRAP ERROR-MSC NOT IN WAIT

Explanation: When read by the @LMS 0 instruction,the MSC status word failed to show that the MSC was busy.

MSG194--IOP INTERRUPT REGS A OR B HAVE STUCK BITS

Explanation: Both interrupt register A and interrupt register B failed to be set to 0.

Contents of Error Information Field (On Page 2 of the GPCIPL Menu): Interrupt register A is shown in R3. Interrupt register A is OR'd with interrupt register B and the result is shown in R4.

MSG195--IOP INTERRUPT REG C HAS STUCK BITS

Explanation: Interrupt register C was not reset to 0. The contents of interrupt register C are shown in R3.

MSG196--IOP PROCESSORS ENABLE TEST FAILED

Explanation: The enable processors PCO command failed to enable all processors.

MSG197--BCE #25 FAILED @STP WRAP TEST

Explanation: After running the @STP instruction, the BCE local store for BCE25 (Test BCE) must contain the following:

A0 = 0001 0000_{16}
B1 = 0000 0000_{16}
C1 = 0003 $7FFF_{16}$

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R5 contains the sum of local store registers A0 + B1 + C1.

MSG198--BCE TRANS/RECV REGS LOAD/READ/DISABL TEST FAILED

Explanation: One or more of the following PC instructions failed.

1. Read MIA transmitter status.
2. Read MIA receiver status.
3. Enable MIA transmitters.
4. Enable MIA receivers.
5. Disable MIA transmitters.
6. Disable MIA receivers.

TABLE 9-I.- Continued

MSG199--IOP SELF TEST MSC PROGRAM COUNTER INCORRECT

Explanation: The MSC program counter was not correct after the IOP self test program was processed.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): The address of the MSC program counter is located in R2.

MSG200--ILLEGAL IOP TERMINATE A DIA BIT 12 = 1

Explanation: The Input Output Terminate A (DIA A bit 12=1) must never be set.

MSG201-->>>GPC POWER REFAIL-PROGRAM/MACHINE WERE RESET

Explanation: There was a second power failure before the powerdown procedure for the first failure was complete. The GPC will reinitialize the entire GPCIPL program.

MSG202-->>>GPC POWER REFAIL-FULL RECOVERY SUCCESSFUL

Explanation: There was a second power failure while the program was processing a previous power transient.

MSG203--GPCIPL RR.VV.PP.II.MM LOADED

Explanation: This is not an error message but an advisory for information only. The message advises the crew that GPCIPL has been loaded.

Where:

RR = Release Number
VV = Version Number
PP = Patch Number (always zero)
II = I Load Set Number (always zero)
MM = Mass Memory Area Number

MSG204--GPC ID CHANGED FROM ## TO %%

Explanation: The GPC identification number has changed since the last update.

MSG205--AT LEAST 1 TESTED BCE DID NOT SET ITS IND_BIT

Explanation: The MSC program counter was incorrect.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the address of the MSC program counter.

TABLE 9-I.- Continued

MSG206--MSC DETECTED AT LEAST 1 TESTED BCE IS STILL BUSY

Explanation: The MSC program counter was not correct.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the address of the program counter.

MSG207--AT LEAST 1 TESTED BCE COULD NOT RESET ITS IND_CTR

Explanation: The MSC program counter was not correct.

Contents of Error Information Field (On Page 2 of GPCIPL Menu): R2 contains the address of the program counter.

MSG208--DCP RR.VV.PP.II.MM LOADED

Explanation: This is not an error warning message. Instead, the message advises that the DEU Control Program, has been successfully loaded.

Where:

RR = Release Number
VV = Version Number
PP = Patch Number (always zero)
II = I Load Set Number (always zero)
MM = Mass Memory Area Number

MSG209--ILLEGAL GPC/CRT ID ENTRY

Explanation: The GPC/CRT keyboard entry is invalid.

MSG210--SELECTED FORMAT WAS NOT IN INITIAL LOAD

Explanation: The critical format selected for display on the CRT is not in DEU core.

MSG211--DESIRED SYSTEM SOFTWARE NOT ALLOCATED ON THE MMU

Explanation: The requested PASS or BFS load is not on the MMU.

MSG212--DEU => ## ERASED

Explanation: The DEU selected for the purge option was erased.

MSG213--ITEM 27 PURGE SELECTION NOT 1-8

Explanation: The Memory Purge option number is not a number between 1 to 8.

MSG214--ITEM # 1-16 AND 27-29 NOT VALID IF PAGE 2 IS UP

Explanation: When Page 2 of the GPCIPL Menu Display is up, the items 1-16 and 27-29 are illegal.

TABLE 9-I.- Continued

MSG215--INVALID PAGE 2 REQUEST -> ITEM # 1-16 OR 27-29 IS UP.

Explanation: When an Item # 1-16 or 27-29 are in progress, it is illegal to request Page 2 of the GPCIPL Menu Display.

MSG216--PURGE OPTION MUST BE SET BEFORE ITEM 28 EXEC

Explanation: Item 27 Purge option must be selected prior to performing an Item 28 EXEC start purge process.

MSG217--ITEM N OR ITEM N <N> ERROR => N MUST BE NUMERIC

Explanation: The item number and/or the item option must be a numeric character.

MSG218--ITEM NUMBER < 1 and > ## NOT SUPPORTED

Explanation: An item number less than one or greater than the number specified in ## are invalid numbers.

MSG219--PURGE OPTION IN PROGRESS ## DESCHEDULED

Explanation: The purge option specified has now been de-scheduled.

MSG220--ILLEGAL KYBD ENTRY WHILE ## IN PROGRESS

Explanation: The Memory Purge option is in progress and only item 29 is legal. An illegal entry was keyed in.

MSG221--DEU KYBD BUFFER CHECKSUM ERROR

Explanation: The checksum computed of the DEU keyboard buffer entries is in error.

MSG222--MASS MEMORY 1 OR 2 NOT SELECTED

Explanation: The mass memory unit 1 or 2 was not selected.

MSG223--DEU => ## IS DE-ASSIGNED

Explanation: The DEU ## selected to be de-assigned was de-assigned.

MSG224--DEU ## IS ASSIGNED TO GPC %%

Explanation: The DEU selected to be assigned to a specific GPC has been done successfully.

MSG225--## DEULOAD CHECKSUM ERROR IN TRANSACTION ##

Exaplanation: A DEU data checksum error occurred for MMU transaction ## of the ## DEULOAD. The ## signify the numeric number.

TABLE 9-I.- Concluded

MSG226--BSL AREA: ## CHECKSUM ERROR IN TRANSACTION ##

Explanation: A BSL data checksum error occurred for MMU transaction ##.

MSG227—## PURLOAD CHECKSUM ERROR IN TRANSACTION ##

Explanation: A purge data checksum error occurred for transaction ##.

MSG228—## PURGE COMPLETE

Explanation: The purge option is annunciated as complete.

MSG229—## PURGE STARTED

Explanation: The purge option XX has begun processing.

MSG230--THIS MSG NOT CALLED BY A SVC

Explanation: This message is not output to the display. It is used internally as an end of the message table marker.

TABLE 9-II.- BSL ERROR MESSAGES

MSG001--UNEXP SVC INT

Explanation: An unexpected SVC interrupt was taken with an invalid interrupt code.

MSG002--UNEXP PROG CHECK

Explanation: An unexpected program check was taken for one of several reasons.

MSG003--UNEXP CLOCK2 INT

Explanation: The low 16-bit hardware binary counter causes a macro interrupt every time the counter is decremented to 0 (maximum of 65.536 ms). This causes a PSW swap only if the high counter (location $00B1_{16}$) is also 0.

MSG004--UNEXP IM INT

Explanation: Interrupt register A was not set when an external 0 interrupt occurred.

MSG005--UNEXP EXT 0 INT

Explanation: The address received by the CPU at the CPU/IOP interface, during the DMA operation, contains a parity error.

MSG006--UNEXP EXT 1 INT

Explanation: Master reset has failed to reset interrupt register D.

MSG007--BFS RR.VV.P.II.MM LOADED_SET MMU SEL SW TO OFF

Explanation: This is not an error warning message but only an advisory message used to provide information. The message advises the crew that the Back-Up Flight System has been loaded.

Where:

RR = Release Number
VV = Version Number
P = Patch Set Number
II = I Load Set Number
MM = Mass Memory Area Number

TABLE 9-II.- Continued

MSG008--BSL RR.VV.P.II.MM LOADED

Explanation: This is not an error warning message but only an advisory message used to provide information. The message advises the crew that the Back-Up System Loader has been loaded into the GPC.

Where:

RR = Release Number
VV = Version Number
P = Patch Set Number
II = I Load Set Number
MM = Mass Memory Area Number

MSG009--UNEXP MACH CHECK

Explanation: An unexpected machine check was taken for one of the following reasons:

1. An address parity error was found in locations 28000_{16} to $33FFF_{16}$.
2. The IOP found a parity error in IOP storage.
3. A parity error was found in CPU storage.
4. A data parity error was found in locations 28000_{16} to $33FFF_{16}$.
5. A CPU ROS parity error was found.
6. The machine check interrupt code was greater than 5 or less than 1.

MSG010--UNEXP EXT 3 INT

Explanation: This interrupt is not used except during GPC testing.

MSG011--UNEXP EXT 4 INT

Explanation: This interrupt is not used except during GPC testing.

MSG012--UNEXP EXT 0 INT-W/D

Explanation: The Watchdog timer has timed out and generated an external 0 interrupt.

TABLE 9-II.- Continued

MSG013--EXT 0 INT-IOP FL

Explanation: The redundancy management (RM) voter logic has detected a failure that affects the operational integrity of the machine.

MSG014--EXT 0 INT-C/M ID

Explanation: An unexpected control monitor idle interrupt has occurred.

MSG015--EXT 0 INT-ROS PE

Explanation: A data parity error has occurred, during transfer from IOP read only storage (ROS).

MSG016--EXT 0 INT-IOP FL

Explanation: The input-output oscillator has stopped.

MSG017--UNEXP IOP PG INT

Explanation: An external 2 IOP programmed interrupt occurred during BSL operation. IOP interrupt register C contained an unexpected value.

MSG018--END-X 0 ERRORS

Explanation: The analysis of the contents of the external interrupt A register is complete and all errors have been annunciated.

MSG019--SVC INT-WILD BR

Explanation: An SVC call with an interrupt code of CF9B was received. The call is annunciated and the program goes back to the interrupted point to continue.

MSG020--DEU BITE ERROR

Explanation: The Built in Test Equipment (BITE) in the DEU has detected an error, and the critical BITE error present bit in the header word of the DEU poll response message has been set. The contents of the headerword is shown in the MCDS BITE MODE field (See Callout No. 10 in Figure 6 and Figure 7). Further information on BITE error and the DEU headerword may be found in the DCP Part I Spec, MG017300, Revision A, dated 1 October 1978, or latest change.

MSG021--NO DEU RESPONSE

Explanation: The DEU did not send a poll response message after it was polled by the GPC.

TABLE 9-II.- Continued

MSG022--BFS COMPUTED CHECKSUM ERROR - XACTION ##

Explanation: The checksum furnished by the MMU was not equal to the computed checksum. The computed checksum is stored in location 'BSLBCSUM'. The ## field will contain the number of the MMU transaction when the error occurred.

MSG023--MMU WILL NOT GO READY

Explanation: The switch selected MMU ready discrete bit was not set. DIA Bit 6 indicates that MMU1 is available for use and Bit 7 indicates that MMU2 is available for use.

MSG024--MMU EXPECTED POSITION ERROR

Explanation: After each position or read command is sent to the MMU, the position of the tape in the MMU is interrogated through a position request command. The warning message results when the position given in the MMU response does not correspond to the expected position.

MSG025--MMU SOURCE SELECT SW POS. ERROR

Explanation. Only one MMU should be selected. The IPL source select switch indicates no MMU is selected or both MMU's are selected.

MSG026--MMU BITE STATUS REG A OR B ERROR

Explanation: The MMU status registers were not set to zero after an MMU status command.

MSG027--MMU I/O ERROR BCE 18 OR 19

Explanation: The no-go status was set for BCE18 or BCE19, where BCE18 corresponds to MMU1 and BCE19 to MMU2.

MSG028--INVALID SVC #

Explanation: An unsupported SVC number was detected. This message will be driven onto the CRT if an illegal instruction occurs ($C6C6_{16}$).

MSG029--IMU CHKPT CHEKSUM ERROR-XACTION ##

Explanation: The checksum furnished by the MMU was not equa l to the computed checksum. The computed checksum is stored in location 'BSLBCSUM'. The ## field will contain the number of the MMU transaction when the error occurred.

TABLE 9-II.- Concluded

MSG030--TFL FORMAT=>%% -COMPUTED CHECKSUM ERROR-XACTN ##

Explanation: The checksum computer for the TFL format does not match the given checksum. The ## field will contain the number of the MMU transaction when the error occurred.

MSG031--TFL FORMAT=> 0102 EXPCTD/IS FORMAT %% =>XACTN ##

Explanation: The format loaded is not the expected TFC format 0102. The format %% and transaction ## fields specify the format and transaction that enters in error.

MSG032--TFL FORMAT=> 0105 EXPCTD/IS FORMAT %% =>XACTN ##

Explanation: The TFC format loaded is not the expected. The format %% and TFL format 0105 transaction ## fields specify the format and transaction that entered incorrectly.

MSG033--THIS MSG MARKS END OF MESSAGES

Explanation: This message is an internal end of message table marker.

1 PASS SYSTEM SERVICES OVERVIEW

SECTION 11
PASS SYSTEM SERVICES OVERVIEW

CONTENTS

Section		Page
11.1	FCOS	11.1-1
11.1.1	Time Management	11.1.1-1
11.1.2	Error Log Reset	11.1.2-1
11.1.3	HISAM Dump Software	11.1.3-1
11.1.3.1	Overview	11.1.3-1
11.1.3.2	Hardware IPL HISAM Dump	11.1.3-1
11.1.3.3	PASS HISAM Dump	11.1.3-2
11.1.3.4	Dump Formats	11.1.3-2
11.1.4	Common and Redundant Set Synchronization	11.1.4-1
11.1.5	G3 Archive	11.1.5-1
11.1.5.1	Background	11.1.5-1
11.1.5.2	G3 Archive Load	11.1.5-1
11.1.5.3	G3 Archive Retrieve	11.1.5-3
11.1.5.4	GPC Initialization/Moding to Run and G3 Archive	11.1.5-4
11.1.5.5	G3 Archive Discrepancy Reports	11.1.5-4
11.2	USER INTERFACE	11.2-1
11.2.1	Uplink Software	11.2.1-1
11.2.1.1	Introduction	11.2.1-1
11.2.1.2	Onboard Command Types	11.2.1-1
11.2.1.3	Two-Stage Command Processing	11.2.1-3
11.2.1.4	GPC Main Memory Read/Writes	11.2.1-7
11.2.1.5	DEU Equivalents	11.2.1-8
11.2.1.6	Variable Downlist	11.2.1-8
11.2.1.7	References	11.2.1-9
11.2.2	Downlink Software	11.2.2-1
11.2.3	Software Dump	11.2.3-1
11.2.3.1	Overview	11.2.3-1
11.2.3.2	GPC Memory Spec	11.2.3-1
11.2.3.3	Dump Downlist	11.2.3-2
11.2.3.4	Dump Formats	11.2.3-3
11.3	SYSTEM CONTROL	11.3-1
11.3.1	ICC Buffer and ICC Sumword Logic	11.3.1-1
11.3.2	OPS Transitions	TBS
11.3.3	GPC Initialization	11.3.3-1
11.3.3.1	IPL Initialization	11.3.3-1
11.3.3.2	Non-IPL, Normal Initialization	11.3.3-4
11.3.3.3	System Initialization	11.3.3-10

SECTION 11
PASS SYSTEM SERVICES OVERVIEW

The Primary Avionics Software System's (PASS) Systems Services (SS) software is divided into three major areas. First, the Flight Computer Operating System (FCOS) is to the General Purpose Computers (GPCs) as MS-DOS or UNIX is to the personal computer (see section 11.1). Second, the user interface (UI) software is needed to handle uplink, downlink, and other computer communication tasks (see section 11.2). Finally, the System Control (SC) software provides for GPC initialization and reconfigurations such as restring (see section 11.3).

The hierarchical structure of the SS software is shown on figure 11-1. This information was taken from the Detailed Design Specification (DDS) produced by International Business Machines (IBM) under contract number NAS 9-14444. The information following the figure applies not only to SS but to PASS and Backup Flight System (BFS) as well. It is included here to give an overview of the software build process.

A. Software Development Process

Both the PASS and BFS undergo the same cycle on the road to certification. This path consists of initiation, initial testing, reviews, detailed testing, more reviews, performance testing, final reviews and certification. Generally, Orbiter software is tested in a bottom-up fashion from individual modules to entire software releases. Moreover, Orbiter software is phase-tested to ensure that it functions properly in all flight phases. Below is a general timeline for a typical software release. Refer to the example of Operational Increment (OI)-8A milestone charts and generic development flow charts for a more graphic view of the software development process (figs. 11-2 through 11-10).

1. Initial software development (8 months)

a. Operational increment baseline process (4 months)

The preparation to begin the development of a new Space Shuttle System operational increment (OI) software release consists of two phases. First, 1 month is required to develop the proposed OI content definition. This process consists of the engineers in the user community developing a list of Software Change Requests (SCRs) which are necessary or desirable for the Space Shuttle Program. This list will generally contain new SCRs and SCRs which have been in work during previous OI releases but have never been approved. Moreover, requirement waivers and discrepancy reports which have been previously dispositioned may be corrected in the OI release. An SCR submittal cutoff date is established by the Shuttle Avionics Software Control Board (SASCB). After this date, only changes which are requirements

to continue space shuttle operations will be admitted into the baselining process.

After the SCR cutoff date, approximately 3 months are needed to baseline the SCRs for the new OI release. First, the list of SCRs is prioritized by the technical community. The Flight Operations Integration Group (FOIG), chaired by the Assistant Director for Space Shuttle Operations, approves the prioritized list of SCRs for the Mission Operations Directorate (MOD). MOD's prioritized list is then integrated with other user group's lists by the Space Shuttle Software Engineering Office. The resulting integrated, prioritized list is reviewed by the SASCB. The list is then reviewed by the software development contractors to determine which SCRs can be implemented with the available resources. Finally, the SCRs which will actually be implemented in the OI, are approved by the SASCB. The resulting baseline of SCRs for the OI is reviewed by the Program Requirements Control Board (PRCB) which is chaired by the Space Shuttle Program Deputy Director. At this point, the OI is ready for development.

b. Level 1 - Module Test

Consists of open loop, standalone testing to verify a single unit of code. A unit is a segment with a single entry and exit, and not larger than a High order Algorithmic Language for Shuttle (HAL/S) compilation module. Level 1 testing is performed during the code development cycle against a prebuild floor load module. The modules of code are compiled and run until there are no errors. Specifically, equations, logic paths, and variable ranges are tested at this level.

c. Level 2 - Function/Interface Test

Consists of closed loop testing to verify interfaces and system integrity after major capability CR's have been implemented. Level 2 testing is usually performed after the code development cycle against a floor release load module prior to First Article Configuration Inspection (FACI). In level 2 testing, individual HAL/S modules are run on the IBM mainframe host under a controlled environment. Specifically, the module interfaces and the user commands are tested at this level.

d. Level 3 - Multiple Module Test

Level 3 testing is similar to level 2 testing with the exception that multiple modules of HAL/S code are compiled and run together. Specifically, functional interfaces, multiple functions, and timing are tested at this level.

2. System build and integration (4 months)

a. Level 4 - System Test

Level 4 testing is post-build testing with a full prereleased software package. These tests are run in the Software Production Facility (SPF) using Flight Equipment Interface Devices (FEID). System interfaces and the mission profile are tested at this level. Control logic interfaces, operational sequence transitions, major mode transitions, and display and control processing are all tested at this level.

b. Level 5 - Detailed System Test

More detailed, full-up testing of different requirements. Mass Memory Unit (MMU) utilization and system level tests are performed at this level.

c. TCT - Test Configuration Team Meeting

The purpose of this meeting is to review specifications for performance testing on new capabilities incorporated into each new OI release. The objective is to baseline a set of performance tests which have thorough technical community review and approval. These tests include Guidance, Navigation, and Control (GNC) software, System Software (SSW), and Vehicle Utility (VU) software.

d. FACI - First Article Configuration Inspection

This meeting is held to review the CRs, DRs, Program Waiver's, memory usage, Central Processing Unit (CPU) duty cycle, etc. associated with an OI release for level 7 verification; i.e., SPF tests, only. Finally, the FACI will officially release an S/W load.

3. System verification (8 months)

a. Level 6 Testing - Requirements Test

This level of testing ensures that all of the software requirements are satisfied. Moreover, timing interfaces and detailed/functional tests are performed.

b. Level 7 Testing - Performance and Acceptance Tests

These are the tests which were reviewed at the TCT and released at the FACI. Specific CRs incorporated in GNC application software are phase-tested to ensure that they perform properly in their designated phases. Also generic phase cases are run to test systems software. Tests verify CPU operation under stress and in generic operation such as OPS transitions. Finally, tests are run to examine and verify operation of VU software. All level 7 testing is performed in the SPF on the third floor of building 30 in the mission support wing.

c. PTR - Preliminary Test Review Meeting

This meeting reviews the results of the level 7 testing. Software anomalies are discussed and open items are addressed.

d. CI - Certification Inspection

This is the final review and certification of a software release before it is released for field use and to the reconfiguration process. This inspection also reviews the results of tests performed on code dealing with GMEMs, patches, discrepancies, and User Notes.

BFS OI-8A MILESTONES

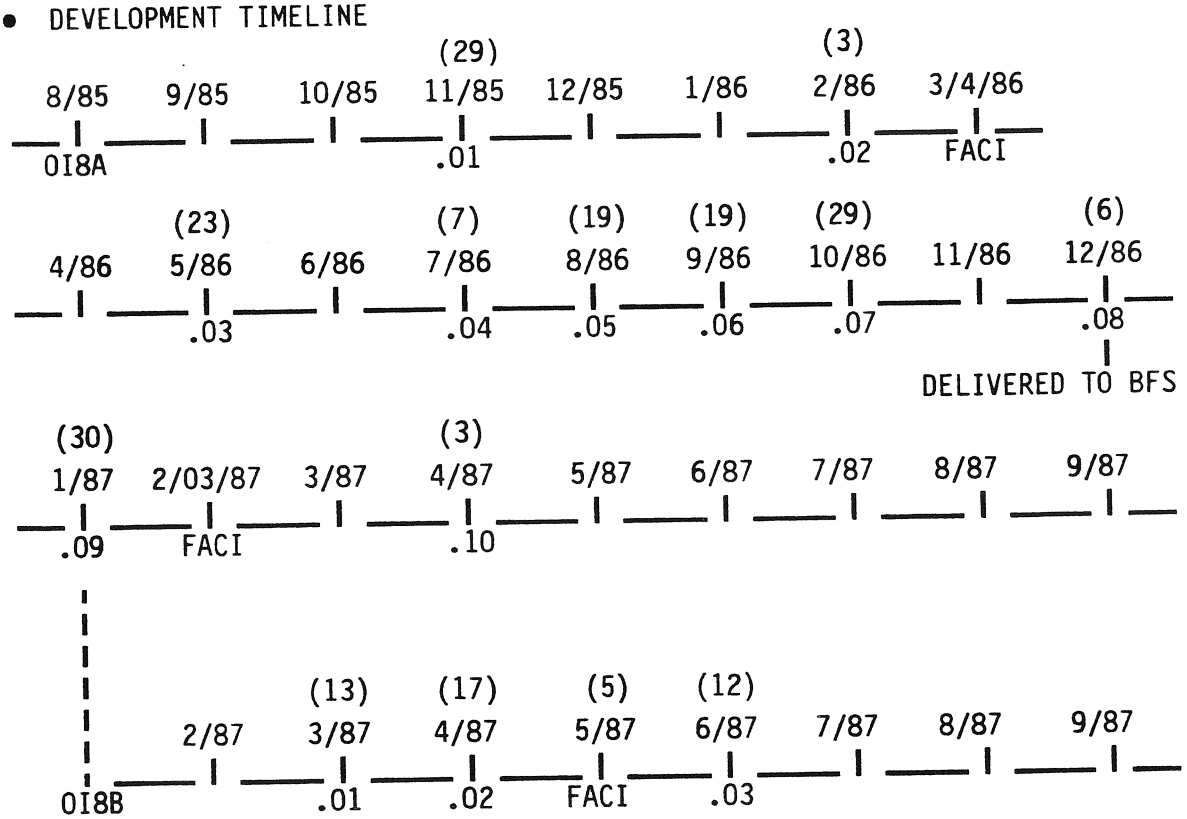
MILESTONE	1986									1987											
	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC
OI-8A																					
• BUILD NO. 1 (8.0)		△ 28																			
• SASCB BASELINE						△ 2															
• BUILD NO. 2 (8.1)								△ 13													
• BUILD NO. 3 (8.2)									△ 19												
• TCT/STM REVIEW										△ 21											
• FACI											△ 3										
• VERIFICATION COMPLETE															△ 5						
• RELEASE TO RSOC															△ 5						
• PTR																△ 26					
• CI																△ 2					
• RSOC RELEASE																		△ 2			

JSC-18820*015

Figure 11-2.- Example of Shuttle Software milestones.

OI8A DEVELOPMENT HISTORY

- BASE SYSTEM WAS OI7C.13
- OI8A BASELINED AUGUST, 1985 (OI8A.01 - OI8A.03
- 1ST OI8A FACI MARCH 4, 1986
- OI8A CR APPROVAL ACTIVITY STARTED IN MAY, 1986
- 2ND OI8A BASELINE IN SEPTEMBER OF 1986
  (OI8.04 - OI8A.08) FLIGHT SAFETY CRS
- OI8A.09 ADDED TO BASELINE TO ACCOMMODATE CR79964F
  RCS REG FAIL PROTECT SEQ.
- DEVELOPMENT TIMELINE



- 490 UNIQUE MODULES WERE CHANGED/ADDED

JSC-18820*016

Figure 11-3.- Example of Shuttle Software Development Timeline.

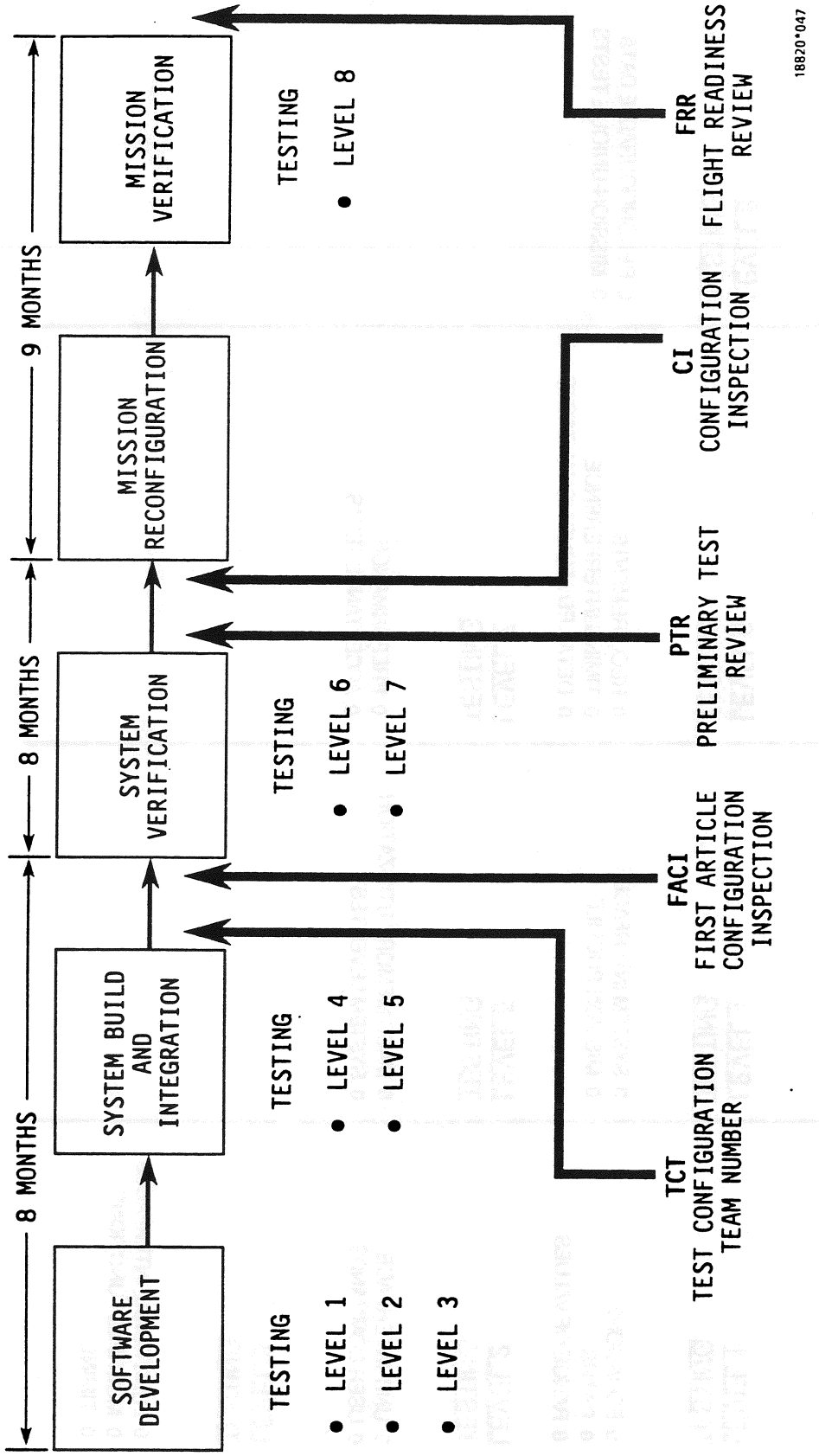


Figure 11-4.- Pass Software Process overview.

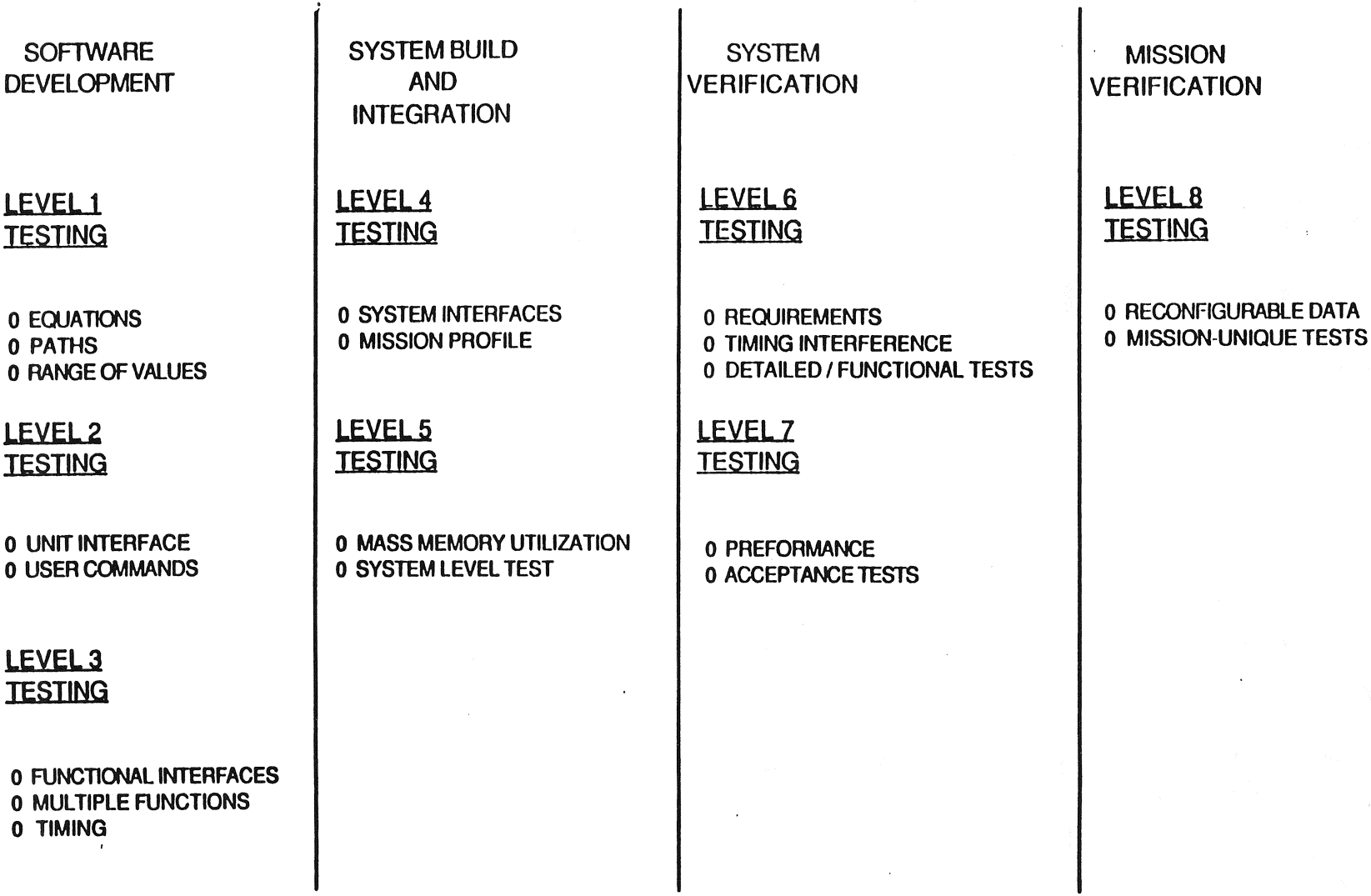


Figure 11-5.- PASS Integrated Test Approach.

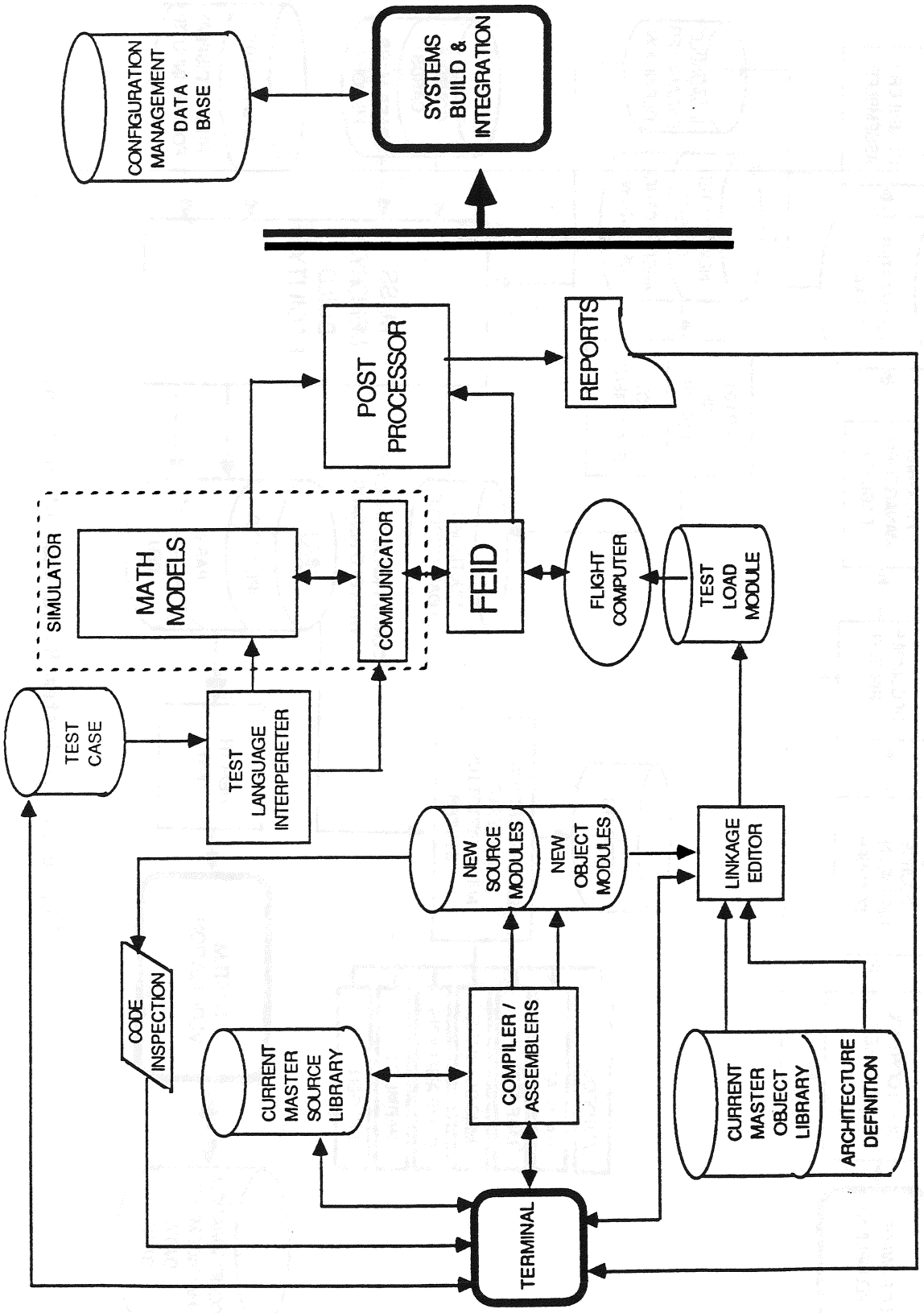


Figure 11-6.- Software Development (FSW/SDF).

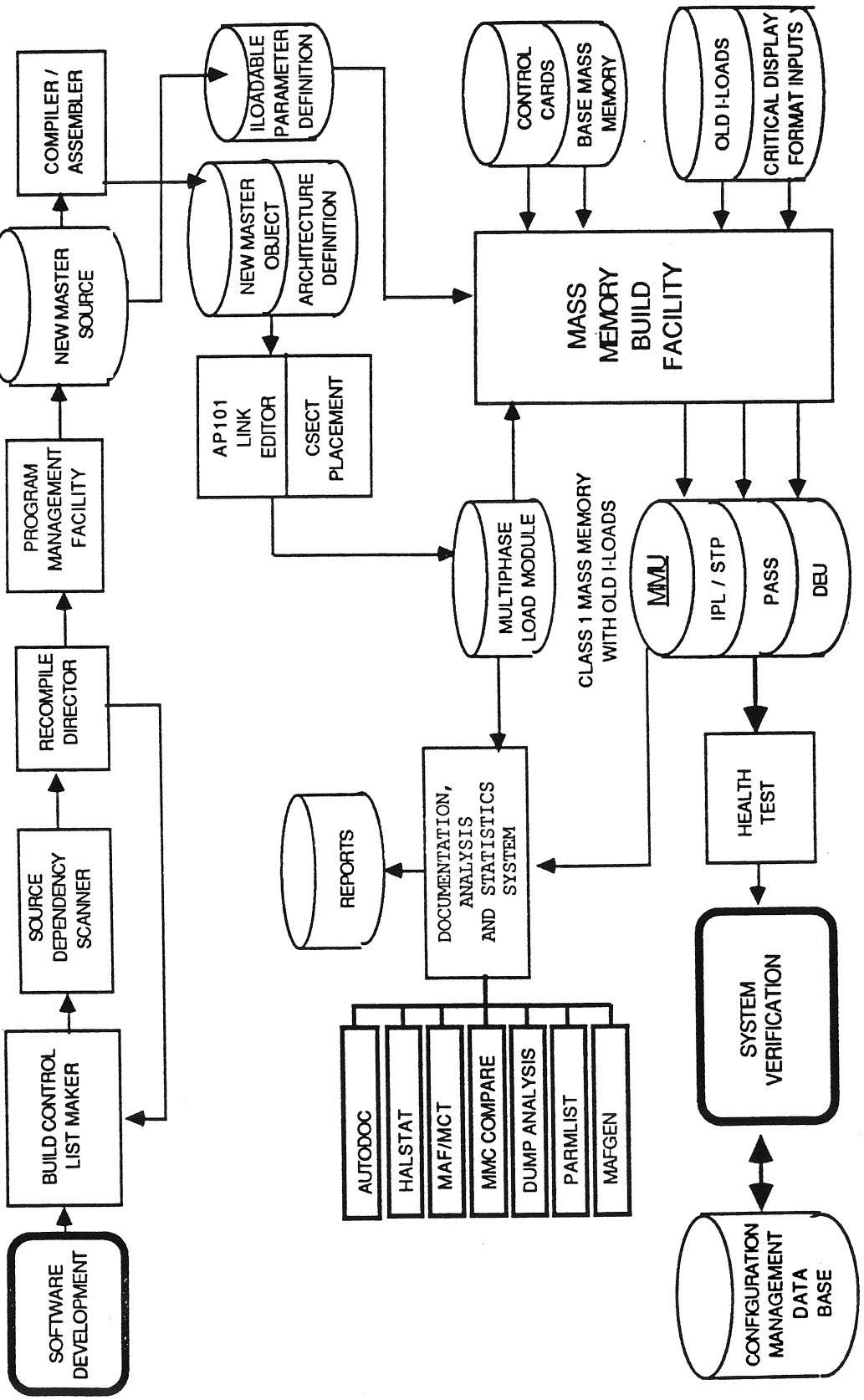


Figure 11-7.- System Build and Integration.

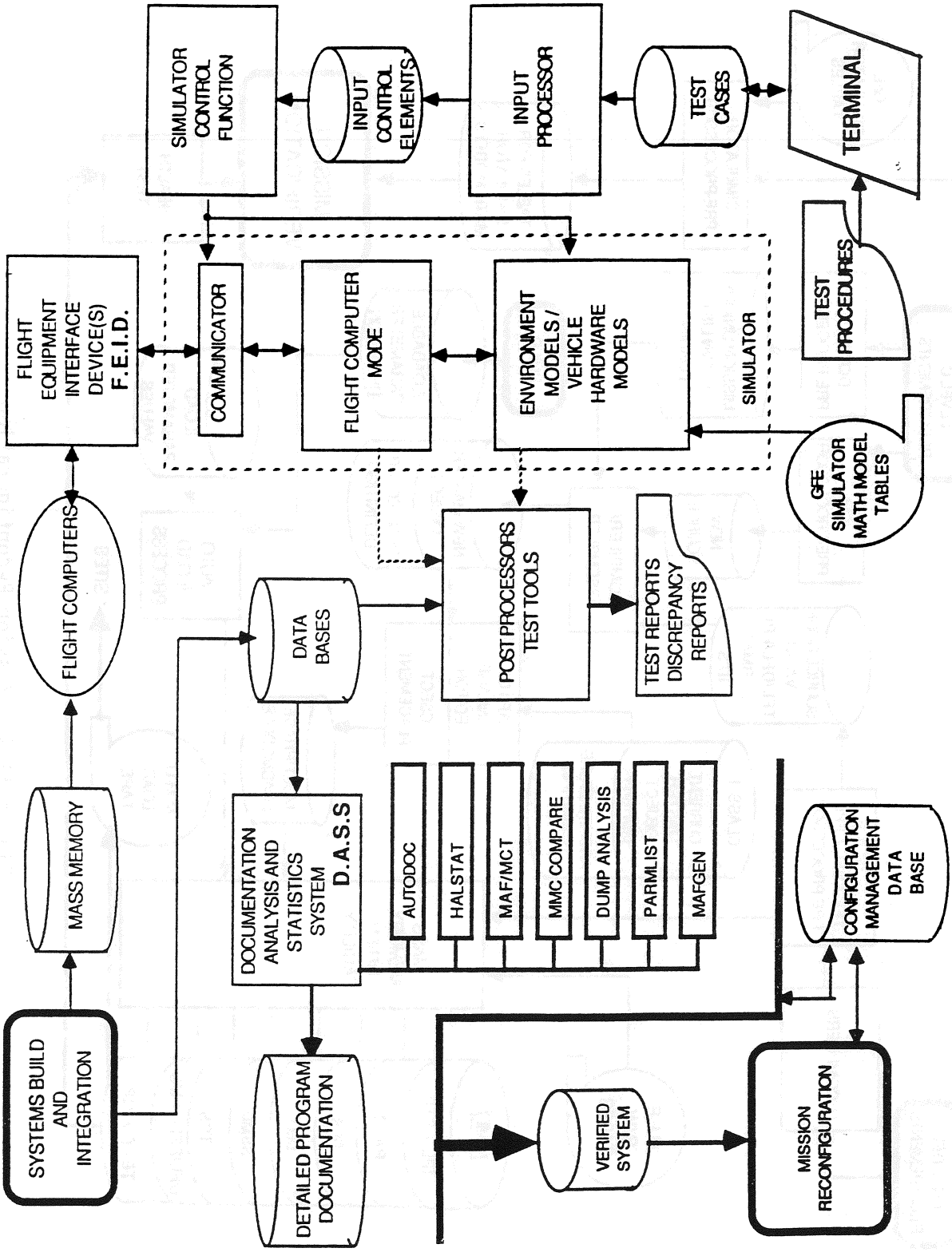


Figure 11-8.- System Verification.

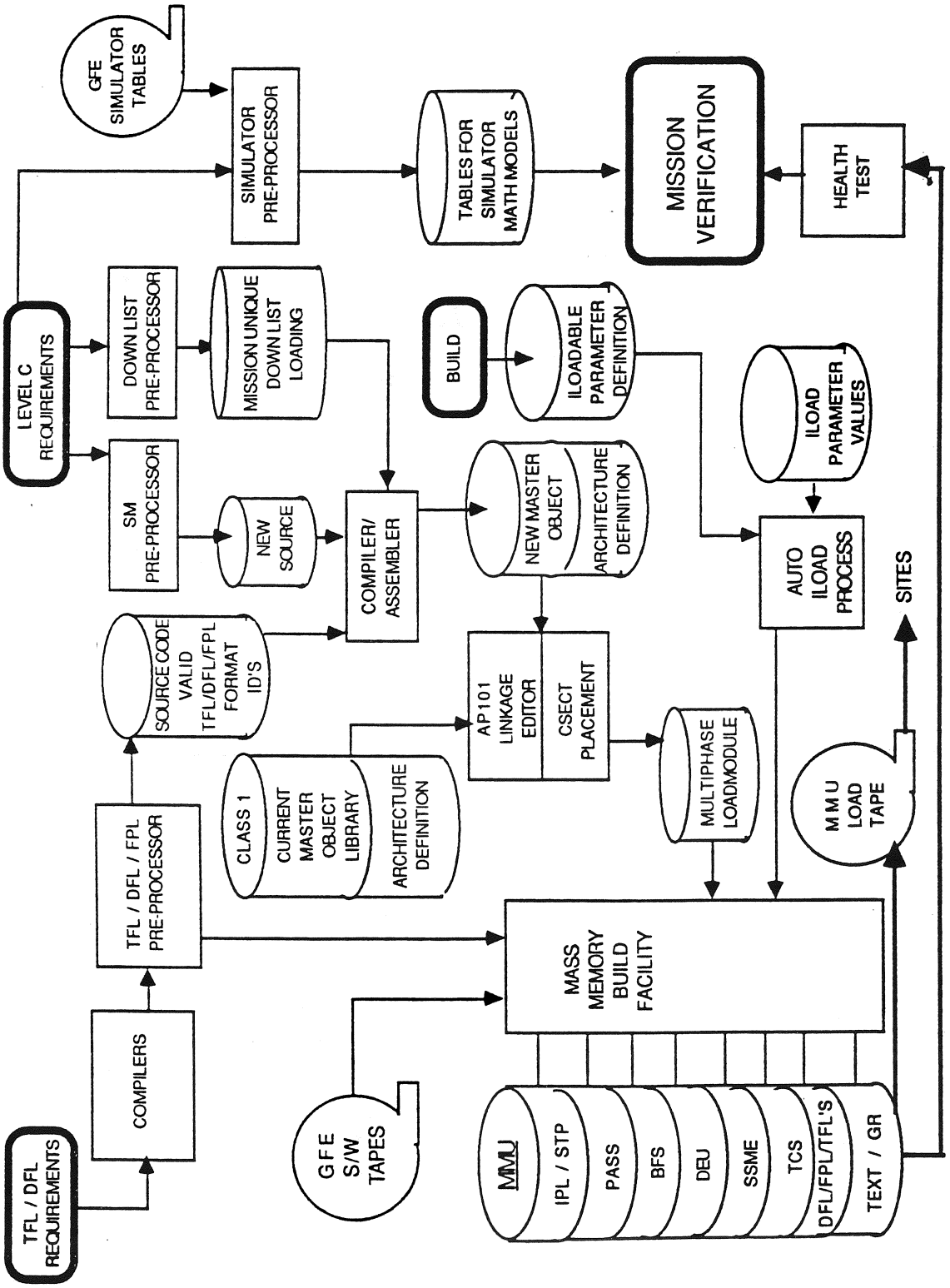


Figure 11-9.- Mission Reconfiguration.

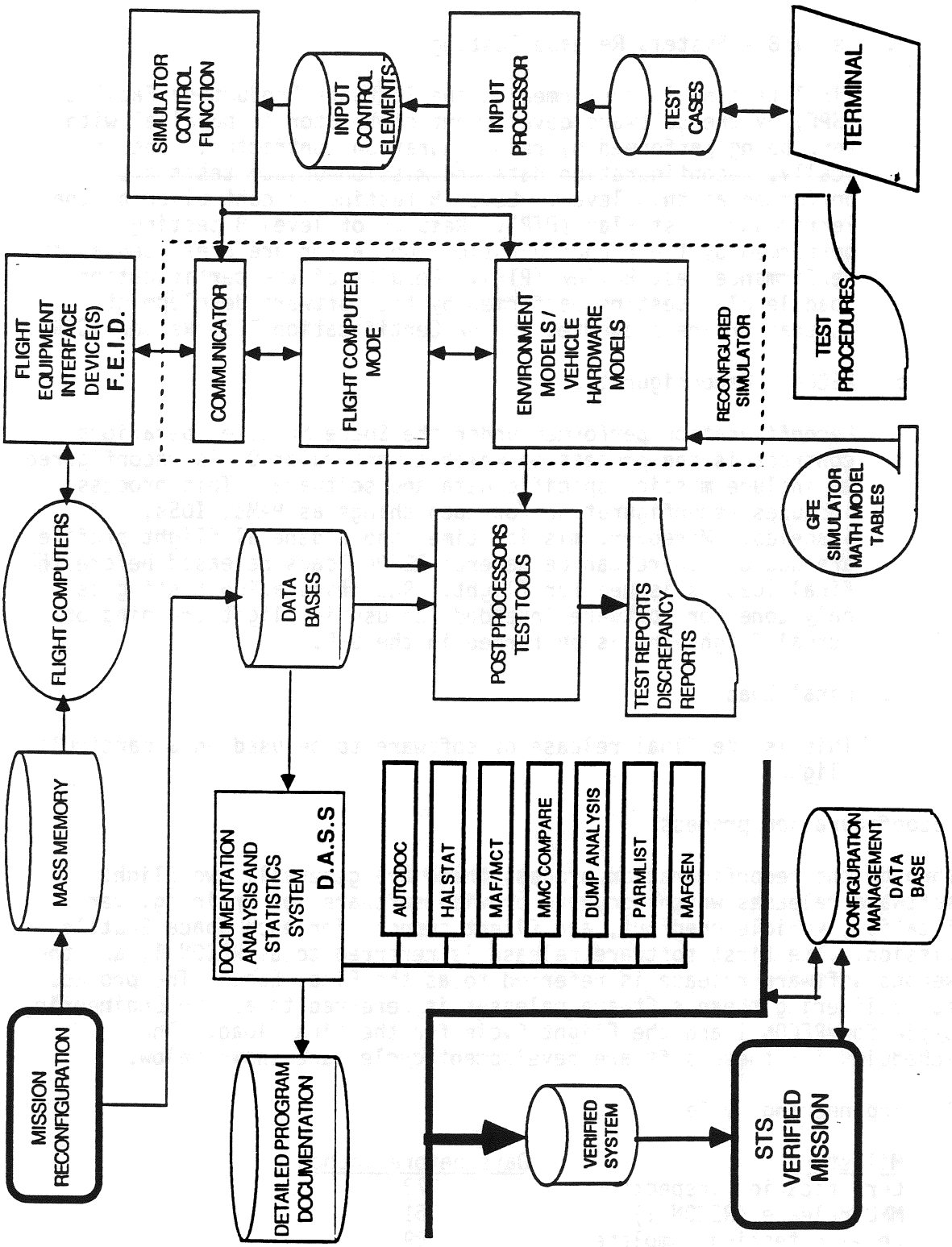


Figure 11-10.- Mission Verification.

4. Mission verification (9 months)

   a. Level 8 - Systems Release Testing

      Final testing is performed in the Software Production Facility (SPF) by the software development contractor in parallel with work being performed by reconfiguration contractor. Specifically, reconfiguration data and mission-unique tests are performed at this level. Level 8 testing is controlled by the Performance Test Plan (PTP). Results of level 8 testing performed by the reconfiguration contractor are presented at the Performance Test Review (PTR). Results of the certification load level 8 testing performed by the software development contractor are presented at the Certification Test Review (CTR).

   b. RECON - Reconfiguration

      Reconfiguration performed under the Space Shuttle Operations contract is the process by which a particular OI is reconfigured to include mission specific data and software. This process includes reconfiguration for such things as PAMs, IUSs, and Spacelab. Moreover, mission times and a general flight profile are added. There can be several RECON loads released before the final load is issued for flight. Reconfiguration testing is only done for software intended for use in flight training or actual flight and is performed in the SPF.

   c. Final Load

      This is the final release of software to be used on a particular flight.

B. Reconfiguration process

Through the reconfiguration process there are generally two flight software releases which occur to provide software for training, verification, vehicle checkout, and flight support for each space Shuttle mission. The first software release is referred to as RECON 1, and the second software release is referred to as the Final Load. The process for delivering these software releases is referred to as the Engineering Cycle for RECON 1 and the Flight Cycle for the final load. The schedules for these software development cycles are shown below.

1. Engineering Cycle

Milestone	Days before launch
Certification inspection	273
MMU release (RECON 1)	161
Level 8 testing complete	89

2. Flight Cycle

Milestone	Days before launch
MMU Release (Final Load)	77
Level 8 Testing Complete	28
Certification Test Review	28
Preliminary Test Review	28

A Complementary Load will be produced for flight if there are any patches to the Final Load.

The reconfiguration process consists of integrating the mission-specific payload data and the reconfigurable systems management data into the operational incrementation baseline software. The mission specific data is level C data and is documented in flight specific Level C SM and Payload Flight Software Requirements. The Data Reconfiguration Office (NASA, JSC, DP) creates the level C data for payloads and the vehicle and enters the data into the STAR and MAST Systems.

The reconfiguration process is shown in graphical detail in figure 11-11. The following text describes each section of figures. The description will focus on the inputs and outputs for each tool used in the reconfiguration process. Finally, the final, end user products will be discussed.

1. Automated Reconfiguration Tools

STAR - Space Transportation Automated Reconfiguration System

Inputs
- Payload inputs
- I-load inputs (input by Flight Design, Engineering, and other I-load owners)
- Mission Definition Data Change Request (DCR)

Outputs
- PDS - Payload Data Set (command and telemetry information for payloads)
- PASS RDGS - Reconfigurable Display Generating System
- BFS RDGS - Reconfigurable Display Generating System
- IDS - I-load Data Sets

MAST2 - Measurement and Stimulus System

Inputs
- Vehicle inputs
- PASS downlist loading
- BFS downlist loading

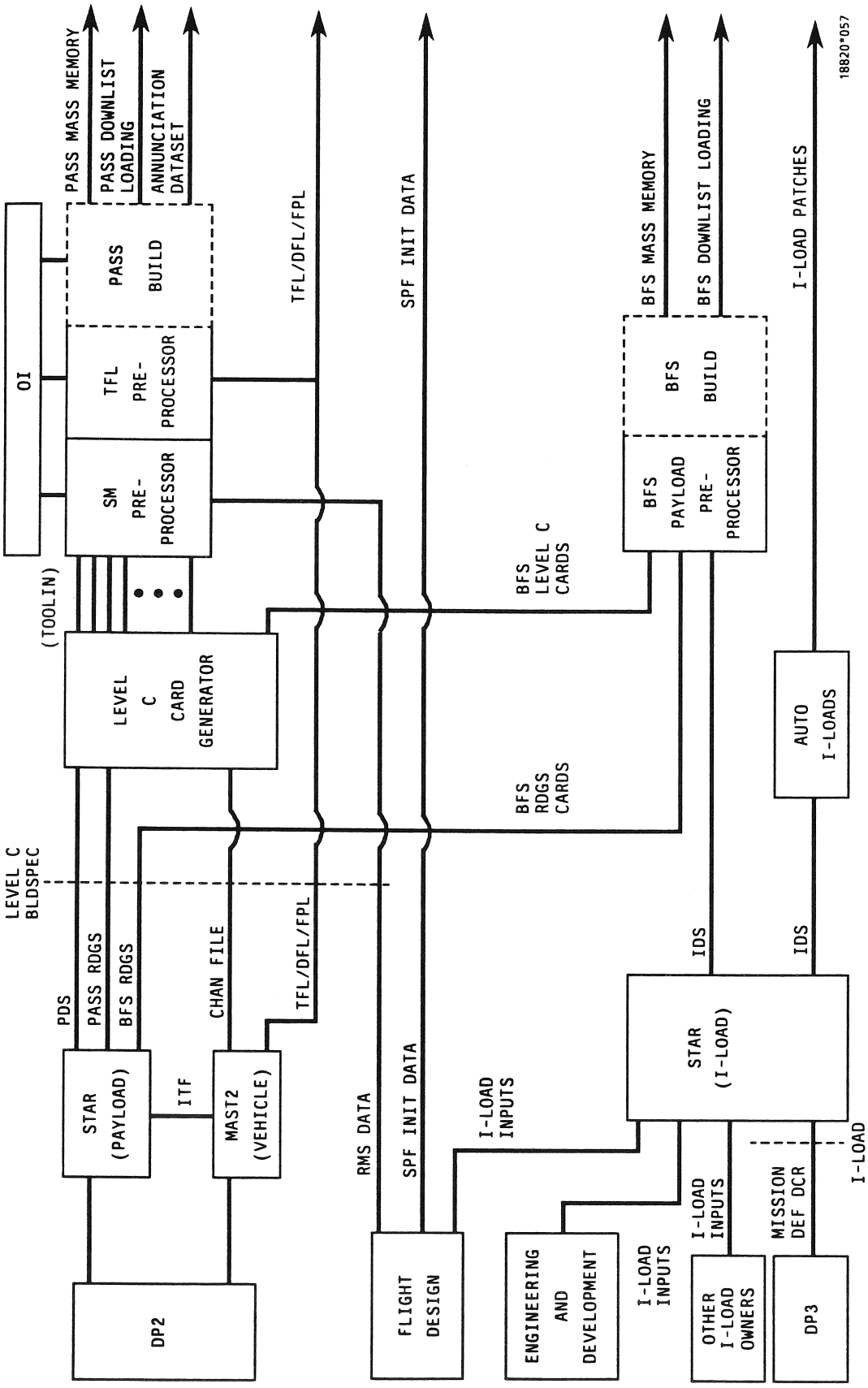


Figure 11-11.- Software reconfiguration process.

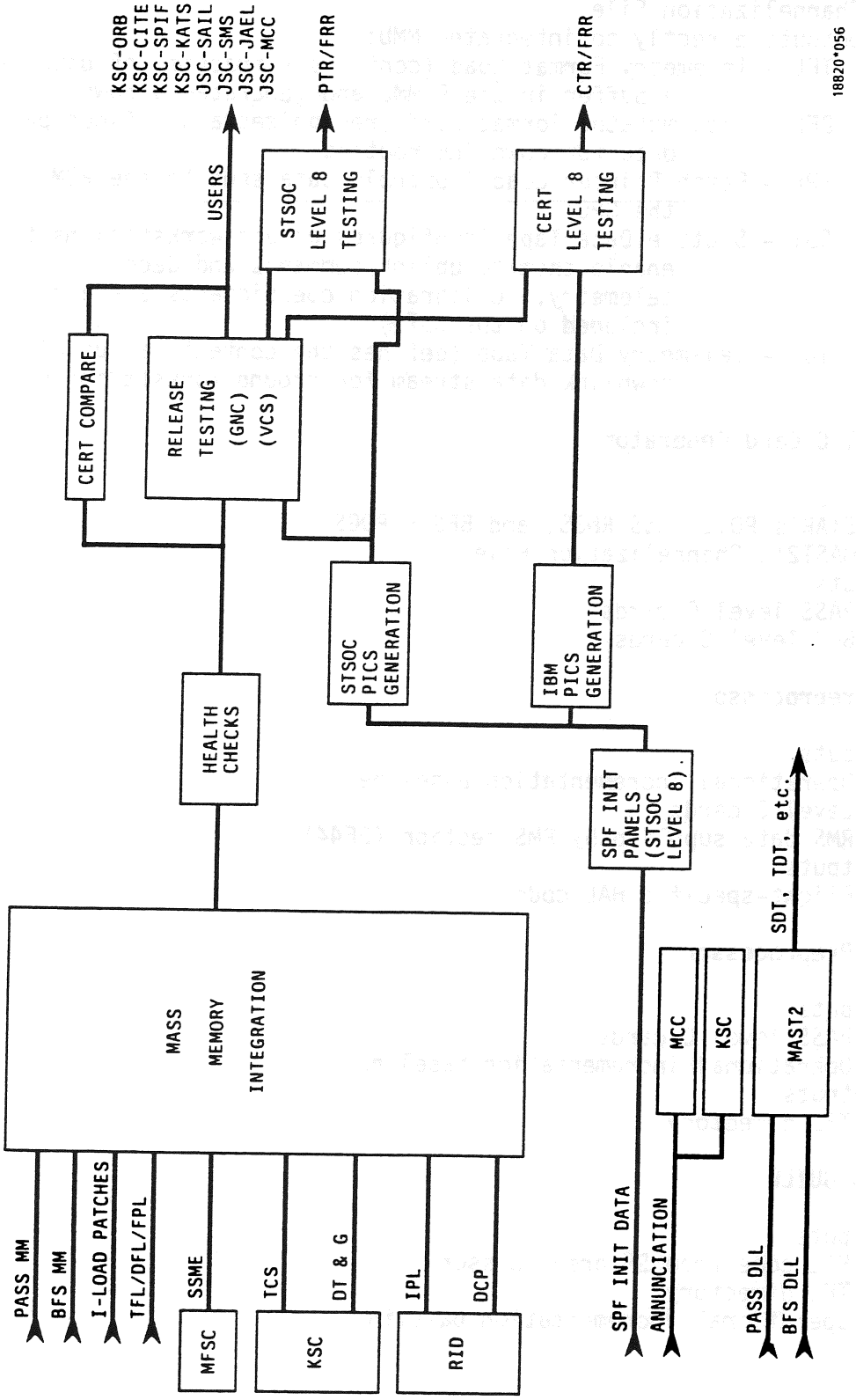


Figure 11-11.- Concluded.

Outputs
Channelization File
Outputs directly to integrated MMU:
TFL - Telemetry Format Load (controls extraction of data from a buffer in the PCMMU and generates a downlink)
DFL - Decommutator Format Load (recognizes and defines payload data for downlink routing)
FPL - Fetch Pointer Load (controls data sent to the PCMMU from the GPC)
SDT - Shuttle Data Tape (configures ground workstations to enable them to uplink commands and decode telemetry. Calibration coefficients are also included on the SDT.)
TDT - Telemetry Data Tape (defines the content layout of the downlink data stream for ground workstation use)

Level C Card Generator

Inputs
STAR's PDT, PASS RDGS, and BFS's RDGS
MAST2's Channelization File
Outputs
PASS level C cards
BFS level C cards

SM Preprocessor

Inputs
Operational incrementation baseline
Level C cards
RMS data supplied by RMS section (DF44)
Outputs
Flight-specific HAL code

TFL Preprocessor

Inputs
PASS level C cards
Operational incrementation baseline
Outputs
TFL directory

PASS BUILD

Inputs
HAL code from SM preprocessor
TFL directory
Operational incrementation baseline

Outputs
PASS mass memory load
PASS downlist loading (shows how the data is organized in a telemetry stream)
Annunciation data set

BFS Payload Preprocessor/BFS Build

Inputs
BFS level C cards
BFS RDGS cards
IDS - I-load Data Sets
Outputs
BFS mass memory
BFS downlist loading

Auto I-Loads

Input
IDS - I-load Data Sets
Output
I-load patches (created for MMU integration)

SPF initialization Panels and STSOC/IBM Predefined Input Control Sequence (PICS) Generation

Input
SPF initialization data
Outputs
SPF simulator loads

2. End User Products

Integrated Mass Memory

Inputs
JSC
Pass MM build
BFS MM build
I-load patches
TFL
DFL
FPL
MSFC
SSME - Space Shuttle Main Engine software
KSC
TCS - Test Control Sequences
DT&G - Display Text and Graphics
Rockwell International/Downey
IPL - Initial Program Load
DCP - Display Electronics Unit Control Program
Outputs

KSC Users
Orbiter
CITE - Cargo Integration Test Equipment
SPIF - Shuttle Payload Integration Facility
KATS - Kennedy Avionics Test Set
JSC Users
SAIL - Shuttle Avionics Integration Laboratory
SMS - Shuttle Mission Simulator
JAEL - JSC Avionics Equipment Lab
MCC - Mission Control Center

Annunciation Data Sets/Shuttle Data Tape/Telemetry Data Tape

JSC MCC
KSC LCC

3. Reconfiguration Testing

Health checks (to make sure that the MMU loads properly to the GPCs)
Release Testing (GNC and VCS; less exhaustive than level 8 testing; ascent and entry case are run in the SPF; vehicle cargo system testing tests SM< FDA, displays, and downlist.)
Certification compare (between STSOC created MMU load and certification load built by IBM
STSOC level 8 testing
IBM certification level 8 testing

C. Definitions

1. I-LOAD

Initial loads (I-LOADs) are values assigned to Orbiter software variables for each RECON and FINAL release of software for a specific mission.

2. K-LOAD

K-LOADs are values given to Orbiter software variables for each OI release. K-LOADs are completely separate variables from I-LOADs.

3. Operational Increment (OI)

An approved/formal release of PASS/BFS Orbiter software. Each OI release updates a previous release with all of the change requests submitted and approved for that OI. PASS and BFS OI release numbers always coincide. K-LOADs change for each OI boundary. I-LOADs change at and within an OI boundary. With approval of the SASCB, each OI release is provided to the community for laboratory testing, crew training, vehicle checkout and flight, and payload checkout and flight. The community includes all development and test facilities listed below.

D. Software control boards and working groups

1. SASCB

The controlling body for changes to Orbiter software.

2. Flight Load Preparation System (FLPS)

An automated system which generates release schedules for the software reconfigurables and submits them with a CR to the SASCB for approval.

3. Orbiter Software Test and Operations (T&O) Board
A management element of the Level 2 SASCB which controls the configuration of formally released software.

4. DR Boards (explanations of these boards TBS)

Development Discrepancy Review Board (DDRB)
Integrated Discrepancy Review Board (IDRB)
Software Discrepancy Management Board (SDMB)
Reconfiguration Discrepancy Review Board (RDRB)

5. Integrated Baseline Control Board (IBCB) (explanation TBS).

E. Orbiter software configuration control documentation

1. Software Authorization Sheet (SAS)

Submitted to the T&O Board secretary to request patches to Shuttle flight software. A SAS must include items such as effect to GPC configuration and MMU tape areas affected by the patch.

2. Release Authorization for Shuttle Software (RASS)

Provides authorization for all unique or early patch applications and is signed by the chairman of the SASCB or his representative. The RASS designates the release as either a "base" or "deviation" patch. A RASS is assigned an Integrated System Number (ISN) for tracking purposes.

F. Program Notes

1. Ops Notes

Ops notes always describe unusual behavior of software that is attributable to a violation of the software requirements. Because the software will be flown with this problem, a waiver that "waives" the requirement always accompanies the ops note. The contractor developing the software either fixes the software, as appropriate, or proposes modifications to the requirements that make them compatible with the code. If the requirement is changed, the ops note may

become a user note and the waiver is deleted. Shuttle Software Division (SSD) publishes the ops notes in the appropriate Program Notes and Waivers (PNW) document. The ops note and waiver are deleted if the software is fixed. Representatives from Mission Operations Directorate (MOD) and Flight Control Operations Directorate (FCOD) inform the proper parties and forward data whenever modifications to flight procedures are necessary.

2. User Notes

A user note clarifies the software operations of code that meets requirements. These notes are published in either the Program Notes and Waivers document and/or the Flight Software User's Guide. Before an OI is implemented, a review determines if CR implementations have changed the software sufficiently to warrant revising or eliminating a user note. Since no requirements violation exists in the software described by the note, there is no corresponding waiver. User notes are eliminated when a CR is approved and the software is adequately changed.

3. Release Notes

Release notes are similar to Ops Notes because the former often (but not always) describe unusual behavior of software due to a requirements violation. However, they differ from Ops Notes by addressing problems with flight software that will be corrected before it is used on a flight. A SAS patch can be used to eliminate these problems in software used by the test facilities. However, software used for a flight is fixed with a CR or a DR. Release notes are issued against software releases containing the known problem. The contractor developing the software fixes the problems addressed within a release note before it is ever used on a flight. T&O board representatives may also request a fix on an earlier OI release via a SAS.

4. Requirement Waiver

Attached to an Ops Note to waive a known requirements discrepancy in a software release.

5. Release Authorization Documentation (RAD)

Authorizes PASS GMEMs for MCC/flight read/write procedures for each flight's clean (non-patched) load.

6. Engineering Order (EO)

Authorizes BFS GMEMs for MCC/flight read/write procedures for each flight's clean (non-patched) load.

7. Orbiter Avionics Mass Memory Computer Program Integration Plan (MIP)

   Controlling document for the responsibilities, release control, and tape formats required to support the building and delivery of Space Shuttle flight software deliverable products. The MIP defines the procedures and responsibilities used to assure correct content and maintenance of these products.

G. Software Development and TEST Facilities

Shuttle Avionics Integration Laboratory (SAIL) at JSC
Shuttle Mission Simulator (SMS) at JSC
Guidance and Navigation Simulator (GNS) at JSC
Orbiter Vehicle at KSC
Cargo Integration Test Equipment (CITE) at KSC
Kennedy Avionics Test Set (KATS) at KSC
Shuttle Payload Integration Facility (SPIF) at KSC
JSC Avionics Equipment Lab (JAEL) at JSC
Software Development Facility (SDF) at JSC
Software Processing Facility (SPF) at JSC

H. Major Areas of Orbiter Software

Backup System Loader (BSL)
Backup System Service (BSS)
Guidance, Navigation, and Control (GNC)
Sequencing (SEQ)
System Maintenance/System Procedures (SM/SP)
Initial Program Load (IPL)
Space Shuttle Main Engine (SSME) Load
Display Electronics Unit (DEU) Load
Text and Graphics (T&G)
Test Control Sequences (TCS)
Telemetry Format Load (TFL)
Decommutator Format Loader/Format Program Load (DFL/FPL)
Software Avionics Command Support (SACS)
Flight Computer Operating System (FCOS)
User Interface (UI)
System Control (SC)

I. Past and Current Status of Shuttle Software

Once the software has been tested, it is released by the SASCB chairman or his representative to the community which includes laboratory testing, crew training, vehicle checkout and flight, and payload checkout and flight.

OI-1 - Basic space shuttle capability.

- STS-1 through STS-8.

OI-2 - Rendezvous and spacelab capability.

- STS-9, 41-B, 41-C.

OI-3 - Main engine control redesign.

OI-4 - Payload remanifest capabilities.

- STS 41-D, 41-G, 51-A, 51-B, 51-C.

OI-5 - Crew enhancements.

- STS 51-D, 51-F.

OI-6 - Experimental orbit autopilot and enhanced ground checkout capabilities.

- STS 51-G, 51-I, 51-J, 61-A, 61-B

OI-7 - Used on STS 61-C and STS 51-L.

- Used as a baseline for OI-9.0 and OI-8A.

- Included advanced propellant dump software and western test range capabilities.

- STS 61-C, 51-L

OI-7C - Designed for use with Centaur payloads.

- Used as a baseline for OI-9.1.

OI-7D - Used for the 61-MT load.

OI-8A - Originally designed for use on STS 26, first STS re-flight

- Baselined from OI-7.

OI-8B - Used on STS 26, first STS re-flight, through STS-30.

- Baselined from OI-8A to include software which allows activation of a newly designed escape system which could be activated in MM 603 or MM 305 with MACH < 0.95.

OI-8C - Used on STS-31, STS-32, STS-34, and STS-36.

OI-8D - Planned for use on STS-35, STS-37, STS-38, and STS-40.

OI-8F - Originally OI-12. First load for new GPCs. Only operating system changes will be implemented. Planned for use on STS-42 through STS-44.

OI-9 - Designed to be used on both the AP101B, old GPC, and the AP101S/G, a version of the new GPC created for software testing and consisting of a new CPU and an old IOP.

- A few problems were encountered with the differences between the two computers. The runtime library needed to be modified for the AP101S/G. However, the completion time for High Frequency Executive (HFE), max rate for BFS, and Mid Frequency Executive (MFE), alt rate for BFS, processes is approximately 33 percent faster. Hence, 33 percent more time is available for background processes every minor cycle. There were virtually no HFE cycle wraps encountered in GNC processing due to the faster execution speed.

- Centaur software from the OI-7C baseline was removed from the PASS OI-9.1 release only.

OI-9A - Designed for new GPC SAIL tests.

OI-10 - Originally developed for use with the old GPCs but has since faded out of view as a result of the STS down time.

OI-11 - Originally designed to be the first software load to fly new GPC software CRs over and above operating system changes.

- Modified to be AP101S development software only. Will not fly.

OI-12 - Designed for use with the first flight of the new GPCs, the AP101S. Modified to become OI-8F. Moreover, OI-12 was used to help develop the new compiler for the new GPCs.

OI-20 - Designed for use in new GPCs.

- First time major changes will be allowed to new GPC software.

- Baselined from OI-8F.

- Takes the place of OI-11.

- Planned for use on STS-43 through STS-45.

OI-21 - 256k Memory GPC system preliminary.

- Planned for use on STS-49 and subsequent flights.

J. References

1. Onboard Shuttle Software Primary Avionics Software Subsystem Flight History and Statistics as of December 1986, IBM Space Shuttle Programs, 12046HM5.C03-1.

2. Operational Increment Development and Reconfiguration Process, DP/Jefferson D. Powell, January 11, 1990.

3. Space Shuttle Programs Orbiter Avionics Software Operational Detailed Design Specification (DDS), Volume: II - System Services; April 2, 1982; Part 1, 2, and 3.

11.1 FLIGHT COMPUTER OPERATING SYSTEM[1]

A. Introduction

FCOS is the major section of system services software, also just called system software (SSW), that has direct control over the GPC hardware. The other major sections of SSW are User Interface (SB 11.2) and System Control (SB 11.3). FCOS initiates and provides basic hardware control services for system control. FCOS also provides general processing control, input and output, and configuration management services to the user interface, system control, and application software. More details on FCOS can be found in the SSW DDS (after which this brief has been patterned), the SSW Maintenance SPEC, and the software module microfiche listings.

1. Initialization

When power is applied to a GPC, one of the FCOS initialization sequences is performed. First, if the entire GPC is to be reloaded, an IPL is performed after power on in the HALT mode and software is automatically loaded from mass memory by the hardware. The second is a SYSTEM RESET interrupt which corresponds to the normal restart sequence. In this case, FCOS: (1) synchronizes the internal timers with the MTU and (2) initiates the system control function.

FCOS now begins accepting interrupts associated with system control and/or software descendants. After each interrupt, the highest priority process is activated until the GPC is taken to STBY/HALT. At STBY/HALT, the hardware stores the GPC status in main memory to be available when the GPC is taken to run again. At that time, FCOS will be started again with one of the two initialization sequences executed.

2. Processing

FCOS processes all interrupts and returns GPC control to the highest priority process. The only exception to the activation of the highest priority process is when a lower priority process is holding a shared resource required by the highest priority process. The shared resource could be either the execution of an exclusive procedure, use of mass memory, or a data update block being executed by a process that becomes a lower priority at the time of interrupt. If this occurs, the lower priority process is activated until the shared resource is released.

The processing of an interrupt within FCOS cannot be interrupted except by power off, machine check, instruction monitor, program counter 1 and program interrupts. The seven blocks shown in the flow (fig. 11.1-1) represent the prime GPC and the basic types of interrupts. In the failure and processing exception cases, the FCOS performs one of the following actions:

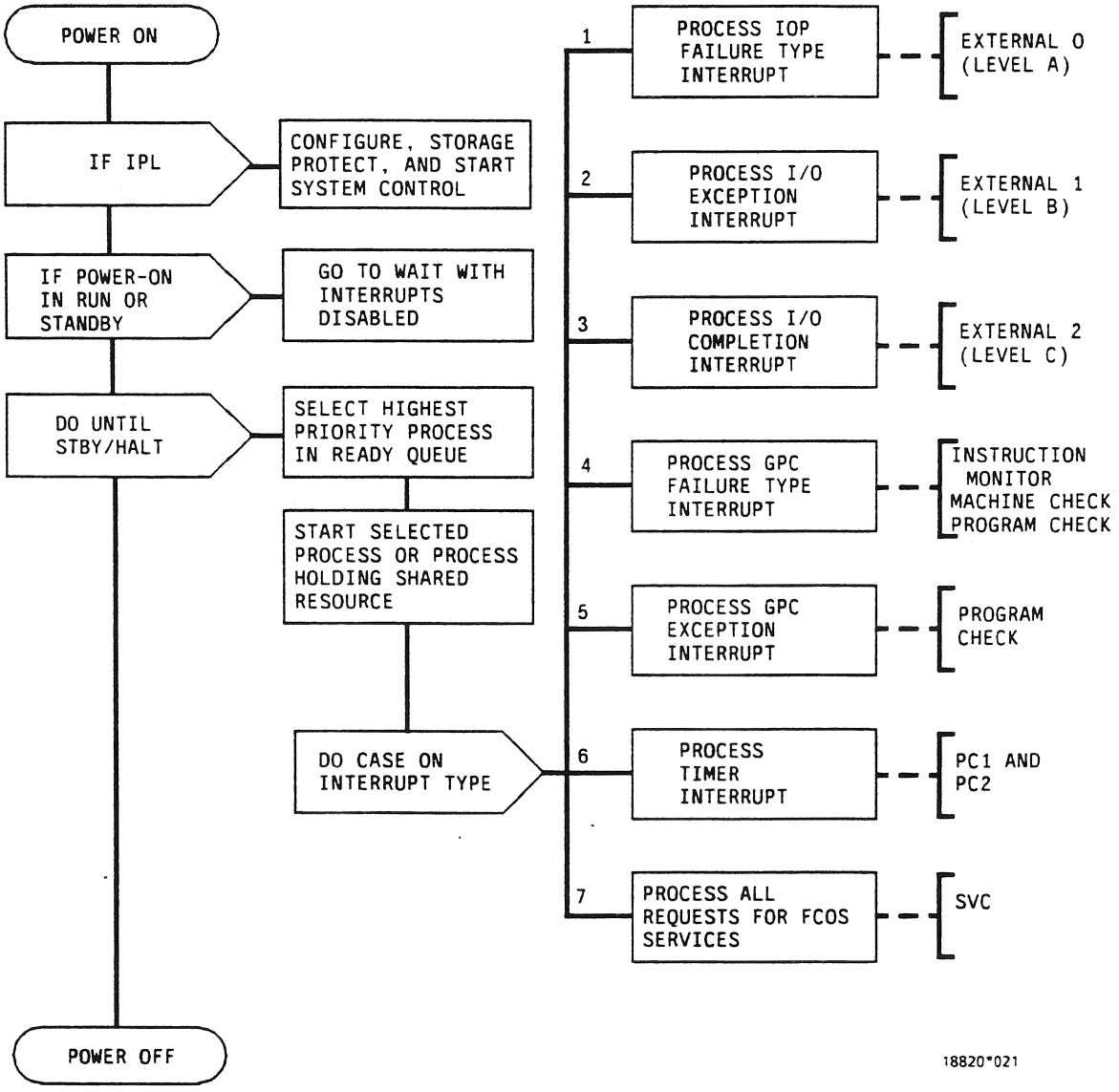


Figure 11.1-1.- Flight computer operating system processing overview.

- Ignores the interrupt and continues
- Sends the indication to the application
- Forces close of the process associated with interrupt

After each interrupt is processed, FCOS again activates the highest priority process on the ready queue. A process may be added to or removed from the ready queue within a service request interrupt routine. The services are defined by the real time statements of HAL/S and other functions of system services.

3. Configuration

System control activates user interface functions which can recognize user inputs for system reconfiguration. If it involves memory overlay within a single GPC or single set of redundant GPCs, user interface issues the request for FCOS to make the change. Otherwise, the request goes to the reconfiguration function in system control.

FCOS acquires the overlay from the specified mass memory or another GPC's memory. Therefore, FCOS contains functions to both request and send requested overlays via the MM/LDB buses. Requests to reconfigure IOP buses are all routed through coordinated by system control.

The remainder of this brief is divided into the three major areas of FCOS (fig. 11.1-2):

a. Process Management - The allocation and control of internal computer resources.

b. I/O Management - The servicing of application process requests for input/output and the control of the initiation of and response to the interfaces of the I/O system hardware.

c. Configuration Management - The FCOS software support required for the control of the GPC configuration changes.

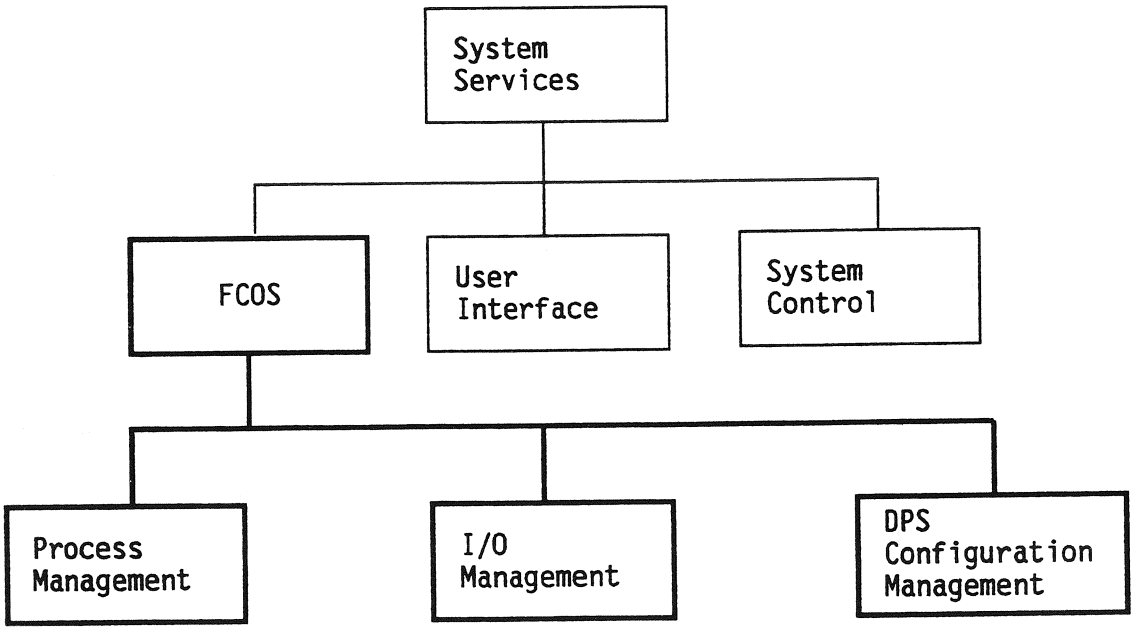


Figure 11.1-2.- SSW Overview Hierarchy Diagram.

B. Process Management

Process management is the control and allocation of computer resources to application software. Process management maintains a process run queue that is ordered by predefined priorities of processes supplied to FCOS at process SCHEDULE time. This queue contains control and status information for all currently scheduled processes and is the basis of the FCOS process switching and dispatching functions. A process switch occurs whenever the active process goes into a wait state or completes (normally or abnormally), or a higher priority process is readied. When an interrupt occurs, after servicing the interrupt, the process dispatching function allocates the CPU to the highest priority ready process.

Process management includes FCOS support for the HAL/S real time statements. Included are time and event management services such as the queuing of time/event dependent requests and the activation of these requests when the time/event occurs. Additional time management functions such as maintenance of software clocks, program counters, GO/NO-GO timer, master timing unit (MTU) and handling of application requests for date/time are also provided.

Also supported is the capability to allow a cyclic process to be initialized such that a predefined I/O operation will be automatically initiated by the FCOS upon receipt of the timer interrupt that readies the process. FCOS process management also includes support for error exception handling. Application error recovery overrides are supported as well as system default response actions when no application overrides exist. Figure 11.1-3 illustrates the elements of process management as follows:

- Process Control - The allocation of the CPU to application processes according to predefined rules.
- Time Management - Software support for the program counters, the GO/NO-GO timer, the Master Timing Unit (MTU support included in this area of FCOS does not include input/output support), the software clocks, and the maintenance and activation of time-dependent requests.
- Event Management - FCOS support for the changing of the status of application event variables and the maintenance and activation of event dependent requests.
- Process Error Management - FCOS support for the handling of error conditions occurring during FCOS/application execution.
- Idle_Time_Processor - A routine which approximates the amount of idle time calculated by the GPC duty cycle.

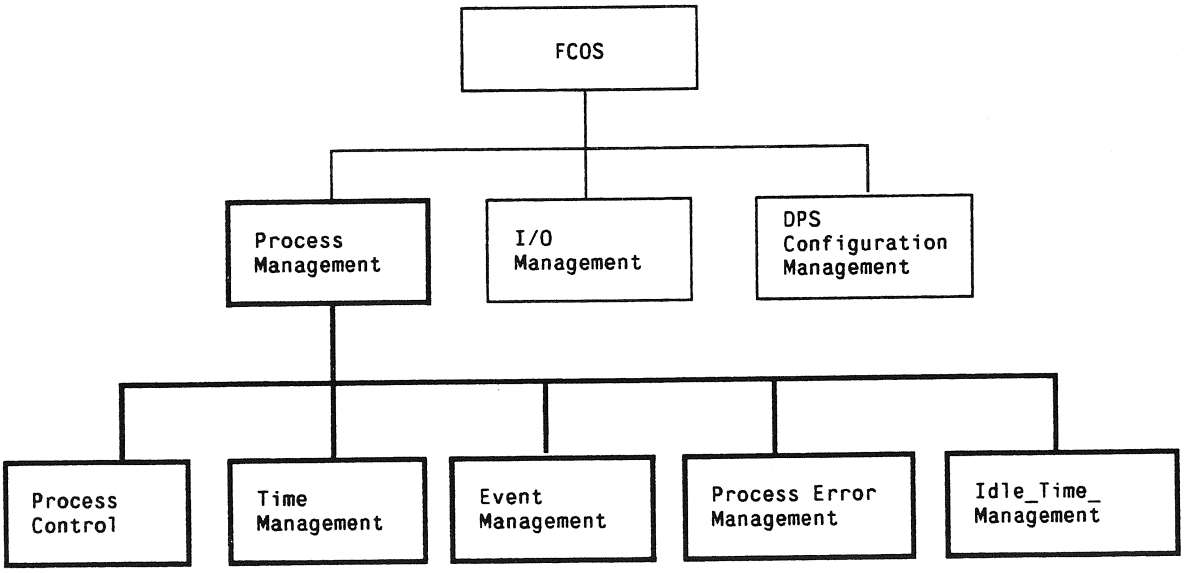


Figure 11.1-3.- Process Management Hierarchy Diagram.

1. Process Control

Process control consists of the interface between the HAL/S application requests and FCOS, the servicing of process control related SVC requests, and the allocation of control of the CPU to application processes.

The application program interfaces with FCOS services via the SVC instruction which causes an interrupt with control being passed to the SVC handler. The SVC handler routes the SVC parameter list associated with the request to the appropriate FCOS routine to service the request. After the SVC request has been serviced, the process dispatcher passes control of the CPU to the highest priority ready process. To indicate to the process dispatching function when a process switch should occur, FCOS routines utilize the process switcher which indicates in an FCOS control block (communications vector table (CVT)) when a process switch is to occur. When a process is readied, process switching is invoked to determine if the readied process is now highest in priority.

The main source of status and control information needed to perform process control functions is the process run queue. The process run queue is a priority-ordered queue of process control tables (PCT's). The ordering is based on preassigned priorities communicated to FCOS at process SCHEDULE time. When a process is scheduled, a PCT is initialized and maintained on the process run queue until the process completes (normally or abnormally). A scheduled process may exist in one of three states:

a. An active state in which the process has control of the CPU.

b. A ready state in which the process is capable of utilizing the CPU but does not have control.

c. A wait state in which the process must have some time pass, an event occur, or its I/O complete before it becomes ready.

A process is said to be within an execution cycle after it has been readied to begin a cycle and before it issues its CLOSE statement to complete the cycle.

The information saved in the PCT comes from two major sources: the SCHEDULE parameter list and the process directory entry (PDE) associated with the process being scheduled. Each process which is to be scheduled must be known to FCOS via a PDE (generated as a control section or CSECT by the HAL/S compiler for each process).

In addition to information in the PCT and the PDE, the FCOS process control functions maintain status information in the CVT. Contents of this table which are of primary importance to process control are

a. A pointer to the beginning of the process run queue

b. A pointer to the PCT of the active process

c. A pointer to the PCT of the next to execute process

FCOS routines manipulate the next-to-execute field to indicate when another process (other than the active process) should be given control. Whenever this next-to-execute field contains a PCT address, this PCT is the highest priority ready process. After an interrupt has been serviced, i.e., its associated process is running, control of the CPU is given to a process other than the interrupted process when:

a. The interrupted process has entered the wait state.

b. The interrupted process has completed (normally or abnormally).

c. The interrupted process is no longer the highest priority ready process.

The remainder of the detailed description of process control is divided into the following major areas:

a. SVC_Handler - Routing of supervisor call requests to appropriate FCOS service routines.

b. Process_Scheduler - Servicing of an application SCHEDULE request.

c. Process_Switcher - Determining whether a readied process is the highest priority ready process.

d. Process_Dispatcher - Passing control of the CPU to the appropriate process.

e. WAIT_Processor - Servicing of an application request to delay processing.

f. UPDATE/EXCLUSIVE_Reserve_Processor and UPDATE/EXCLUSIVE_ Release Processor - FCOS control of resources associated with UPDATE blocks and EXCLUSIVE procedures.

g. CLOSE_Processor - Cleanup and/or reinitialization functions necessary at the end of the current execution cycle of a process.

h. TERMINATE_Processor - Servicing of an application request to terminate itself or other process(es).

i. CANCEL_Processor - Servicing of an application request to cancel itself or other process(es).

j. OPS_CANCEL_Processor - Servicing of an application request to cancel all the processes within a major function except for the specified process(es).

k. Select_I/O_Processor - Servicing of an application request to associate timer initiated I/O with a cyclic process.

l. Chain_PCT_Routine, Release_PCT_Routine, Free_PCT_Routine - PCT handling routines.

m. Disable_Interrupts_Routine, Enable_Interrupts_Routine - Servicing of application requests to disable and enable interrupts to protect shared data variables (FCOS/system control).

n. Zero_Sync_Discrete_Processor - Services requests to put up the zero sync code (000) and keeps it up until a GPC goes to RUN.

o. Reset_Counter_Processor - Resets the normal error counter for a given device.

2. Time Management

Time management includes the FCOS support for all HAL/S real-time statements which have time as a parameter, the FCOS support of MTU updates, MTU/software clock synchronization, the FCOS support of a software backup to the MTU, and the FCOS support of timer initiated I/O. In support of these functions, the FCOS builds table entries called timer queue elements (TQE's). A TQE contains information which identifies its purpose, the time at which it is to expire or a time increment, and the process to which it applies. The description of time management is divided into the following areas (For detailed information see SB 11.2.1):

a. Timer_Queue_Generator - The process of creating the active TQE chain.

b. GPC_Clock_Expiration_Processor - Results of a PC1 interrupt.

c. TQE_Expiration_Processor - Results of a PC2 interrupt.

d. TQE_Dequeue_Processor - Removing expired TQE's from the active TQE chain.

e. Time/Date_Application_Requests_Processor - Supplies time and date to requesting applications.

f. MTU_Update_Processor - FCOS involvement in updates to the MTU.

g. MTU_Redundancy_Manager - Synchronization of MIU and software clocks.

h. Time_Conversion_SVC_Services - Handles conversion SVC's.

i. Convert_To_Fixed_Point_Routine and Convert_To_Floating_Point_Routine - Conversion routines.

j. Current_GMT_Routine - Supplies current GMT.

k. Program_Counter_2_Update_Routine - Updates PC2.

l. Fixed_To_MTU_Format_Conversion_Routine and MTU_To_Fixed_Format_Conversion_Routine - Conversion routines.

m. Chain_TQE_Routine - Handles TQE's.

n. Expiration_Time_Update_Routine - Updates expiration time.

3. Event Management

FCOS event management is responsible for evaluating event conditions in application process SCHEDULE and WAIT statements, for retaining the event conditions for future evaluation if they are not immediately satisfied, and for changing event variable states.

Event management retains event conditions for future evaluation in table elements called event queue elements (EQEs). An EQE contains flag bits to indicate the purpose of the EQE, a pointer to the PCT of the process to which the EQE applies, and the event condition to be evaluated. EQEs which are retained for future evaluation are said to be in the active EQE pool. However, FCOS only supports the following types of event expressions:

a. Expressions containing a single event variable with/without "NOT".

b. Expressions containing more than one event variable connected by all "OR" operators.

c. Expressions containing more than one event variable connected by all "AND" operators.

There are two types of events: normal events and process events. Normal events are declared and controlled by an application process. Process events are created automatically by the HAL/S compiler for every scheduleable process. A process event is true when the process is in the FCOS process run queue and false at all other times.

Whenever a process is scheduled, the FCOS process scheduler sets the process event for the process to TRUE, and the event evaluator is CALLed to evaluate any retained EQEs which reference the event. Whenever a process is removed from the process run queue via a CLOSE, CANCEL, or TERMINATE SVC, or as a result of SCHEDULE cancellation conditions being met, the process event for the process is reset to FALSE, and the event evaluator is CALLED to evaluate any retained EQEs which reference and the affected process event.

Anytime a retained EQE is evaluated and found to be satisfied or the process to which a retained EQE applies is removed from the process run queue, the EQE_Dequeue_Processor is CALLed to remove the EQE from the active EQE pool. Event management software is divided into the following areas:

a. Set_Event_Processor - Sets event to true state.

b. Reset_Event_Processor - Resets event to false state.

c. Signal_Event_Processor - Signals event true.

d. Event_Queue_Generator - The process of adding an EQE to the active EQE pool.

e. Event_Evaluator - Evaluation of EQEs in the active EQE pool.

f. EQE_Dequeue_Processor - Removing a satisfied EQE from the active EQE pool.

4. Process Error Management

Process error management provides FCOS support for the handling of error conditions which occur during FCOS or application execution. If the application programmer has not specified otherwise, FCOS performs system defined default actions when errors occur. These system default actions have no relation to the group assignments but are based on severity of the error and information available to FCOS at time of the error. The FCOS ground rule in performing default actions is always to attempt to continue execution within the constraints which exist at the time of error. System default recovery is not unique for each error but consists of three basic approaches:

a. Ignore the error and continue processing.

b. Force close the application process and continue processing.

c. Discontinue current FCOS processing and dispatch the highest priority ready application process.

FCOS error annunciation is accomplished by setting indicators which are associated with cases in which FCOS needs error annunciation performed. These indicators are interrogated by the system control system interface processor, and appropriate error annunciation is performed.

Application-related errors include instruction monitor interrupts and program interrupts occurring during application execution, and application errors detected by FCOS. The FCOS routines that detect these errors (in the case of error interrupts, this is the FCOS routine which gets control as a result of the interrupt) CALL the Process_Error_Recovery_Processor to perform appropriate actions based on the error environment and the process in error. Event management services, process control services, and the Process_Error_Logger are CALLed as necessary by the Process_Error_Recovery_Processor. This routine returns to the CALLing FCOS routine after the error has been processed.

FCOS errors are handled within the Program_Interrupt_Handler and the instruction monitor interrupt handler. In this case, the process error logger is called directly by these programs. FCOS queue overflow results in a protection interrupt during FCOS execution and is treated as a severe error by the Program_Interrupt_Handler.

The FCOS does not process machine check error interrupts. The machine check new PSW is set to force the GPC into an uninterruptible wait state on the occurrence of a machine check. The remainder of process error management is divided into the major areas:

a. Program_Interrupt_Handler - The fielding of Program Monitor interrupts and error recovery of FCOS errors.

b. Instruction_Monitor_Interrupt_Handler - The fielding of instruction monitor interrupts.

c. Process_Error_Recovery_Processor - The recovery processing of application related errors.

d. Process_Error_Logger - The recording of error conditions which occur during system execution.

e. Forced_CLOSE_Processor - The cleanup processing necessary when a process encounters certain errors conditions.

5. Idle Time Processor

The Idle_Time_Processor receives control when there is no processing to be done and has a priority of zero. It approximates the total amount of idle time which occurs between computations of CPU duty cycle. The idle time processor runs in an infinite loop and is always on the PCT run queue. It receives control through regular processing of the FCOS process dispatcher when no other process is ready to execute.

C. I/O Management

I/O management services all requests for the passage of data and control information between the flight software and the hardware subsystems. I/O management supports the simultaneous servicing of multiple I/O requests based on the priority of the requested I/O and the availability of the required I/O hardware. Queues of I/O queue elements (IOQEs) containing control and status information are maintained by I/O management. These IOQEs contain information such as the number of halfwords which should be transferred, the I/O device which should be addressed, the address of the user's data area, and an indication of which operation should be performed.

I/O management controls the data transfer functions of the master sequence controller (MSC) and the 24 bus control elements (BCEs). I/O management allows a requesting process to wait for its I/O to complete or to proceed concurrent with its I/O operations. I/O control tables are initialized for use by BCE programs to interface with I/O devices.

The service of initiating processing in the MSC is provided for I/O and other FCOS functions. I/O management provides facilities for processing I/O completion interrupts from the MSC. If requested, I/O management initiates time-tagging of input data. If specified by the requesting process, I/O management sets an event upon completion of the requested I/O service and restarts a process if it is waiting for I/O to complete.

I/O management provides facilities for processing IOP hardware error interrupts and for detection and processing of I/O transmission errors. Transaction error status information is provided to the requesting process if specified, and in some cases, I/O errors detected in one GPC are reported to other GPCs. I/O queue cleanup facilities are provided in support of FCOS TERMINATE, CLOSE, and FORCED CLOSE processing. I/O management is divided into the following areas (fig. 11.1-4):

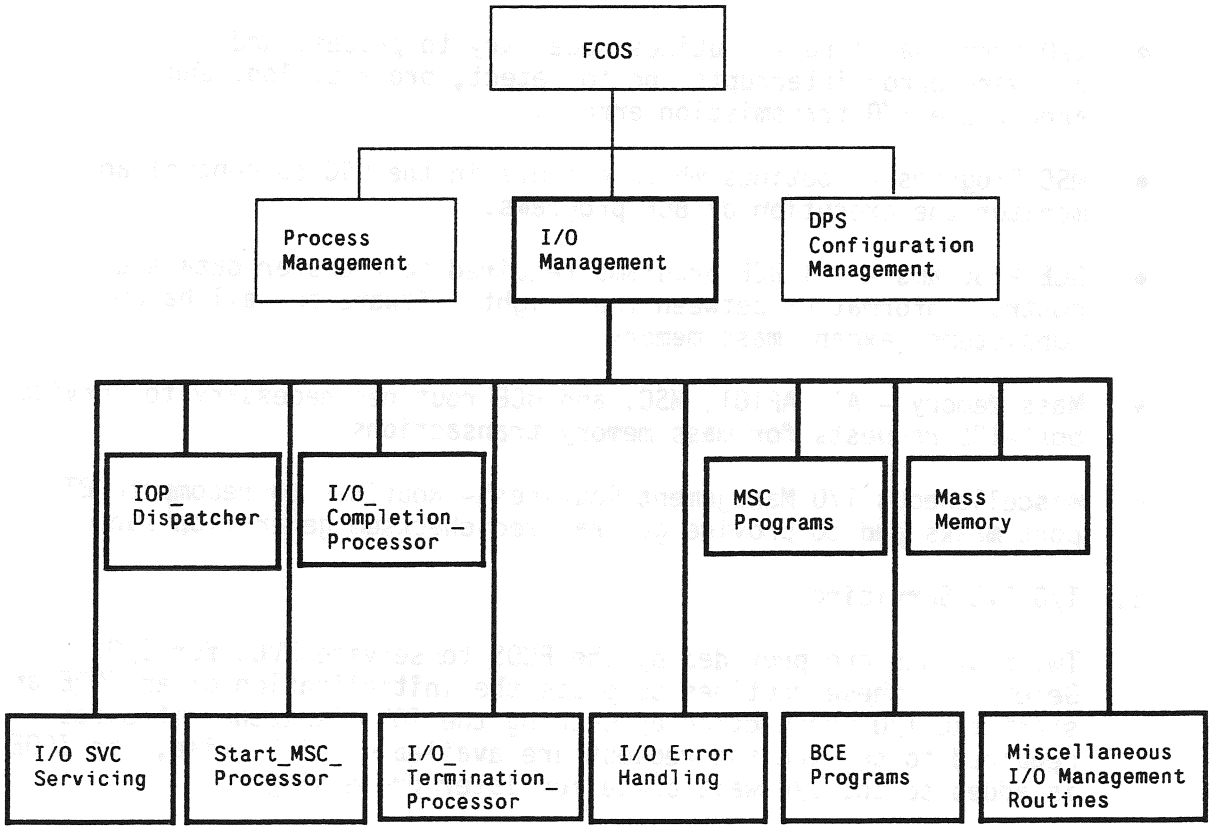


Figure 11.1-4.- I/O Management Hierarchy Diagram.

- I/O SVC Servicing - The processing of incoming request for I/O services.

- IOP_Dispatcher - Routines which modify IOP accessible control tables and causes the initiation of IOP programs to accomplish the requested I/O service.

- Start_MSC_Processor - A routine which initiates MSC processing at the address specified by the calling program.

- I/O_Completion_Processor - A routine which fields I/O completion interrupts from the MSC, restarts processes which are awaiting I/O completion, initiates time-tagging operations, sets I/O completion events, and initiates the reporting of I/O errors to other GPC's.

- I/O_Termination_Processor - A routine which performs I/O queue cleanup in support of FCOS TERMINATE, CLOSE, and FORCED CLOSE processing.

- I/O Error Handling - Routines necessary to process and log IOP hardware error interrupts and to detect, process, log, and annunciate I/O transmission errors.

- MSC Programs - Routines which execute in the MSC to control and monitor the execution of BCE programs.

- BCE Programs - All BCE programs required to transfer data and control information between the flight software and all hardware subsystems (except mass memory).

- Mass Memory - All AP101, MSC, and BCE routines necessary to service post-IPL requests for mass memory transactions.

- Miscellaneous I/O Management Routines - Routines to recompute BTU port masks and to provide generalized checksum generation support.

1. I/O SVC Servicing

   Two routines are provided by the FCOS to service SVCs for I/O. Generally, these routines complete the initialization of an IOQE and start the I/O transaction by CALLing the IOP dispatcher if buses required to service the request are available. Otherwise, the IOQE is added to the I/O wait queue for later processing.

   A generalized I/O SVC service routine gets an unused IOQE from the free pool and initializes it with information from the I/O SVC parameter list. A preinitialized I/O SVC service routine uses preinitialized IOQE's built when the FCOS is assembled. The I/O SVC parameter list for this I/O service routine identifies the preinitialized IOQE needed to service the request. Requests for mass memory services are always passed to the mass memory manager by the

generalized I/O SVC service routine after IOQEs are initialized. I/O SVC servicing is divided into the following areas:

a. I/O SVC Servicing Processor - Generalized I/O SVC service routine which initializes unused IOQEs.

b. Pre-Initialized I/O SVC Processor - I/O SVC service routine which uses preinitialized IOQEs.

2. IOP_Dispatcher

Two routines are provided by the FCOS to dispatch work for the IOP. Generally, these routines do required device dependent processing and starts the transaction by either directly starting the MSC or calling the start MSC processor. The IOP dispatcher places I/O_Queue_Elements on the I/O active queue and initiates the execution of IOP programs to accomplish the I/O requested. The GNC downlist dispatcher performs the same basic function, but is limited to the downlist I/O in a GNC memory configuration. IOP dispatching is divided into the following areas:

a. IOP Dispatcher - Generalized I/O dispatcher which dispatches all I/O, except mass memory I/O.

b. GNC Downlist Dispatcher - A special purpose I/O dispatcher which processes downlist in a GNC environment other than G9.

3. Start_MSC_Processor

The Start_MSC_Processor is CALLed to start the MSC at a program address specified by the CALLing program. If required, it will halt then reenable the MSC prior to loading the MSC program counter (PC) with the specified program address. Also, if required, it will compute for the top active IOQE the time remaining before a MSC timeout is declared. The start MSC processor is CALLed whenever the MSC is required to work. The MSC is normally CALLed to start or monitor an I/O operation.

4. I/O_Completion_Processor

The I/O_Completion_Processor fields and processes all I/O completion interrupts (level C interrupts) from the IOP, recognizes software detected IOP errors and passes control to I/O error management. Checksumming of the variable portion of the ICC buffer is performed on ICC I/O completion. Protected transaction input problem report (IPR) and ICC error retry are initiated from it and will invoke MTU redundancy management processing if a time tag is requested for the completed I/O. Bus reconfiguration processing is invoked if the completed I/O has freed buses needed in reconfiguration processing.

5. I/O_Termination_Processor

The I/O_Termination_Processor dequeues IOQEs from the I/O WAIT queue, and flags IOQEs from the I/O ACTIVE, I/O SUSPEND, and MM ACTIVE queues with Parent_PCT_Terminated status. The flagged I/O ACTIVE and SUSPEND queue IOQEs are dequeued by this routine while the MM ACTIVE IOQE's are dequeued by the MM I/O management routine. Also, this routine flags IOQEs that have invalid PCTs with Invalid_Parent_PCT status in support of FCOS CLOSE, FORCED CLOSE and CANCEL processing.

6. I/O Error Handling

FCOS I/O error handling processes error information received via three types of error interrupts from the IOP:

a. Level A errors (hardware generated)

b. Level B errors (hardware generated)

c. Level C errors (generated under software control in the MSC)

Level C errors are associated with a particular I/O transaction, level A and level B are not. Level A and level B errors, depending on the type of error, are either ignored, or cause the IOP to be reset and the CPU to be put in a disabled WAIT state. Level C errors are indicated to the I/O requestor by means of status flags or COMFAULT status indicators. Status flags warn the I/O requestor that the I/O did not complete successfully. The COMFAULT status indicators warn that data is not available from the indicated LRU's.

All I/O errors are logged. Each GPC maintains a stack of I/O error log entries for the first five I/O errors encountered and a cyclic stack of I/O error log entries for the previous five IOP errors encountered by the GPC. In addition, SSIP maintains in each GPC a copy of the most recent I/O error log entry for each of the GPCs in the common set. A system software error message is annunciated when a BCE element is bypassed. A down arrow display indicator may also be set. The following is a list of I/O error handling programs:

a. Level_A_I/O_Error_Interrupt_Handler - This program processes level A interrupts (hardware generated).

b. Level_B_I/O_Error_Interrupt_Handler - This program processes level B interrupts (hardware generated).

c. Level_C_I/O_Error_Interrupt_Handler - This program processes level C interrupts (generated under software control in the MSC).

7. MSC Programs

The MSC programs are executed by the MSC in the IOP. One of their functions is to load the program counters and start the bus control elements (BCE) which the GPC has designated. One or more buses can be started by the MSC in a single request by the CPU. When the buses are started, the CPU indicates to the MSC which buses are to be monitored (both listen and command mode buses are monitored). The MSC monitors the designated buses and interrupts the CPU upon their completion. It also notes to the CPU whether or not an error has occurred on one of the monitored buses. The GPC stand alone dump is another function performed by MSC code. Note that mass memory I/O is handled by a different set of routines (see mass memory). The following is a list of MSC programs:

a. MSC_Control_Routine - This Program is used to start BCE programs other than mass memory BCE programs.

b. MSC_I/O_Monitor - This program monitors BCE's (other than mass memory) for I/O completion.

c. GPC_Stand_Alone_Dump - This program is used to HISAM dump the contents of GPC memory to downlink when the GPC dump switch has been processed.

d. ICC_Program_Counter_Routine - This program sets the BCE program counters for ICC bus transactions.

e. DEU_Program_Counter_Routine - This program sets the BCE program counters for DEU bus transactions.

f. Payload_Program_Counter_Routine - This program sets the BCE program counters for payload bus transactions.

g. Launch_Program_Counter_Routine - This program sets the BCE program counters for launch data bus transactions.

h. Flight_Critical_Program_Counter_Routine - This program sets the BCE program counters for flight critical bus transactions.

i. PMU_Program_Counter_Routine - This program sets the BCE program counters for PCMMU bus transactions.

8. BCE Programs

Most FCOS BCE programs are designed for a specific bus terminal unit to be used with a specific I/O transaction. However, some BCE programs have been made general enough to be used by several I/O transactions to communicate with different bus terminal units. The BCE programs used with test control supervisor (TCS) I/O requests are examples of generalized BCE programs.

To save CPU setup time, most FCOS BCE programs are hardcoded. This means that the buffer areas used and the data requested by these programs is never changed. Some hardcoded BCE programs are structured to facilitate bypassing of commands to retrieve data from LRU's which are failed or powered off.

A BCE program executes in either the command or listen mode. The BCE program may transmit both commands and data and also receive data while in command mode. In listen mode, the BCE program may receive data but is not allowed to transmit commands or data. A BCE in listen mode relies on BCE programs in other IOP's to command a subsystem to return data. The presence of a command to a subsystem on the listen BCE bus is used as a signal to start receiving data when a receive data instruction is being executed by a listen mode BCE.

For input operations, a BCE program running in command mode executes some delay instructions to allow the listening BCEs to have time to get set up to receive the data. After the delays, the commanding BCE executes a #MIN instruction to command the subsystem to return the appropriate data and then prepares itself to handle this data. When the subsystem places its data on the bus, the command and listen mode BCE's are prepared to accept it, and consequently receive exactly the same data at exactly the same time. The following is a list of BCE programs:

a. DDU_BCE_Processor - This program outputs data to the dedicated displays.

b. DEU_BCE_Processor - This program is used to communicate with all DEU's.

c. ICC_BCE_Processor - This program provides support for SSIP ICC data transfers between common set GPC's and input problem report (IPR) ICC data transfers between redundant set GPC's.

d. LDB_BCE_Processor - This BCE program provides GPC and ground communication via the GSE.

e. MDM_BCE_Processor - This program includes support for MDM BITE acquisition.

f. Payload_Discrete_BCE_Processor - This BCE program supports caution and warning discrete outputs to the payload MDM's.

g. PMU_BCE_Processor - This BCE program provides all FCOS I/O support for the PCMMU.

h. Permanently_Resident_BCE_Processor - This BCE program provides support for MTU reads and writes, TCS I/O request, PCMMU OI/PL data, RAM read, and PCMMU GPC data RAM writes.

i. DL/DEU_Error_Counter - This module clears DEU and downlist error counts after a successful DEU read or downlist write.

j. HFE_Processor - This BCE program provides support for all high frequency executive I/O.

k. MFE_Processor - This BCE program provides support for all midfrequency executive I/O.

l. SM_Fixed_BCE_Processor - This BCE program provides output support for SM PF MDM I/O for payload bay doors, full cell purge, antenna, H_2O loop, and hydraulic fluid.

m. SM_Mission_Dependent_BCE_Processor - This BCE program provides support for mission dependent payload I/O.

n. SRB_BCE_Processor - This BCE programs read data from the SRB LLO1, LLO2, LRO1, and LRO2 MDM's.

o. IMU_BCE_Processor - This BCE program provides support for the IMU I/O.

p. NSP_BCE_Processor - This BCE program reads the network signal processor data.

q. GN&C_Initialization_Input_BCE_Processor - This BCE program reads discretes from the FA and FF MDM's.

r. SM_PF_Fixed_Input_BCE Processor - This BCE program reads the PF MDM fixed inputs.

s. MEC_BCE_Processor - This BCE program reads MEC1 and MEC2.

t. G9_Oneshot_BCE_Processor - This BCE program writes required initialization data to the backup flight system (BFS).

u. MICU_BCE_Processor - This BCE program provides all FCOS I/O support for the RMS MCIU.

v. MTU_PMU_BCE_Processor - This BCE program reads the MTU via the PCMMU.

w. PDI_BCE_Processor - This BCE program reads and writes to the PDI.

x. PSP_BCE_Processor - This BCE program reads and writes to PSP1 and PSP2.

y. NWS_BCE_Processor - This BCE program reads the nosewheel steering position feedbacks.

9. Mass Memory

Mass memory units (MMUs) require special I/O management services due to data verification and hardware characteristics, such a slow access time and fixed length data blocks. These special I/O management services provide MM service request processing, IOP dispatching, completion processing, error processing, and special IOP programs.

MM service requests are initiated using the generalized I/O SVC services. The MM user requests services by issuing an I/O SVC with the associated parameters initialized. These parameters include the MM operation code used to indicate what function is to be performed, the number of words to be transferred, the requested MM starting block address, and other parameters common to all I/O processing. The I/O SVC service routines when invoked will allocate and initialize an IOQE for the request then relinquish request processing to the special MM I/O management services. These services will process the MM service request and monitor the completion of the transaction. If the requested MMU is busy, the services will reject the request as in error.

It should be noted, that data is written to MM in 512 16-bit words regardless of the size of the output buffer or word count specified by the requester. However, the IOP is not required to receive MM data in 512 word blocks. Therefore, reads of less than one block are supported.

MM data verification on read operations is performed by calculating a checksum word from the input data words and comparing the checksum word against the last input data words. The checksum word is calculated by summing the input data words. The last word of any logical data words before the data block was originally written to MM.

Data verification on write operations is performed by rereading the data written to MM and performing a read operation data verification. The data is read into a buffer other than the users output buffer to preserve data integrity. Mass memory I/O management services consist of the following programs:

a. MM/GTG_I/O_Request Processor - Fields all normal mass memory/GPC-to-GPC I/O service requests.

b. MM/GTG_I/O_Dispatcher - Processes all mass memory/GPC-to-GPC I/O services.

c. MM/GTG_I/O_Service_Request_Monitor - Monitors all outstanding mass memory/GPC-to-GPC I/O service request.

d. MM/GTG Builder/Scheduler - Builds and schedules timer queue elements for timing out mass memory/GPC-to-GPC I/O service requests.

e. MM/GTG_I/O_Completion_Processor - Performs mass memory/GPC-to-GPC I/O service request completion processing.

f. MM/GTG_Checksum_Processor - Performs mass memory/GPC-to-GPC data/load block checksum generation and verification.

g. MM/GTG_Error_And/Or_Sync_Processor - Performs mass memory/GPC-to-GPC I/O service request error processing and synchronization processing.

h. MM/GTG/BCE_Error_Processor - Performs mass memory/GPC-to-GPC BCE/MSC error detection.

i. Mass_Memory_MSC_Processor - Initiates MM BCE programs.

j. Mass_Memory_BCE_Processor - MM BCE program used to read, write or position a MMU tape or to read a MM status register.

k. Mass_Memory_Utility_Write_BCE_Processor - MM BCE program used to write 1 to 32 data blocks to a MM.

10. Miscellaneous I/O Management Routines

The miscellaneous I/O management routines are provided for use by FCOS (as well as applications) to perform common I/O-related processing. Currently, there is only one such routine:

Checksum_Generator - Computes a checksum for an indicated buffer.

D. DPS Configuration Management

DPS configuration management is the FCOS software support required for the control of GPC and IOP configuration changes. This includes control of both the internal configuration of a single GPC as well as control over the configuration of the DPS.

Configuration management provides control and sequencing of the GPC initialization. This control includes the loading and initialization of system software into a GPC from mass memory, IOP bus configuration, and verification of the GPC health. Configuration management is responsible for GPC synchronization with respect to time, I/O activity, and data exchange. Configuration management controls configuration changes resulting from internally detected hardware failures. Software directive, or user input.

In providing control of individual GPC main memory changes, DPS configuration management provides control of memory overlays to

predefine memory locations as a result of OPS transitions. The ability to modify main memory or mass memory locations as a result of user input or uplink requests is also provided.

IOP bus configuration changes resulting from hardware failures, sync failures, user requests, or OPS changes are also supported
Figure 11.1-5 illustrates the following elements of DPS configuration management

- GPC Initialization - The loading of system software into the GPC from mass memory, initialization of the IOP, and initialization of the GPC.
- Memory Management - FCOS support for overlays, program modification, and user modification of BCE chains.
- Miscellaneous_CM_Request_Processor - FCOS support for SVC requests to read or write discretes, set/reset the CAM lights, and set/reset the termination control latches.
- Bus Configuration - The control of IOP bus configuration changes resulting from hardware failures, sync failures, user requests, or OPS changes.
- GPC Redundancy Management - Provides for GPC synchronization with respect to time, I/O activity and data exchange, and also provides for fault detection.

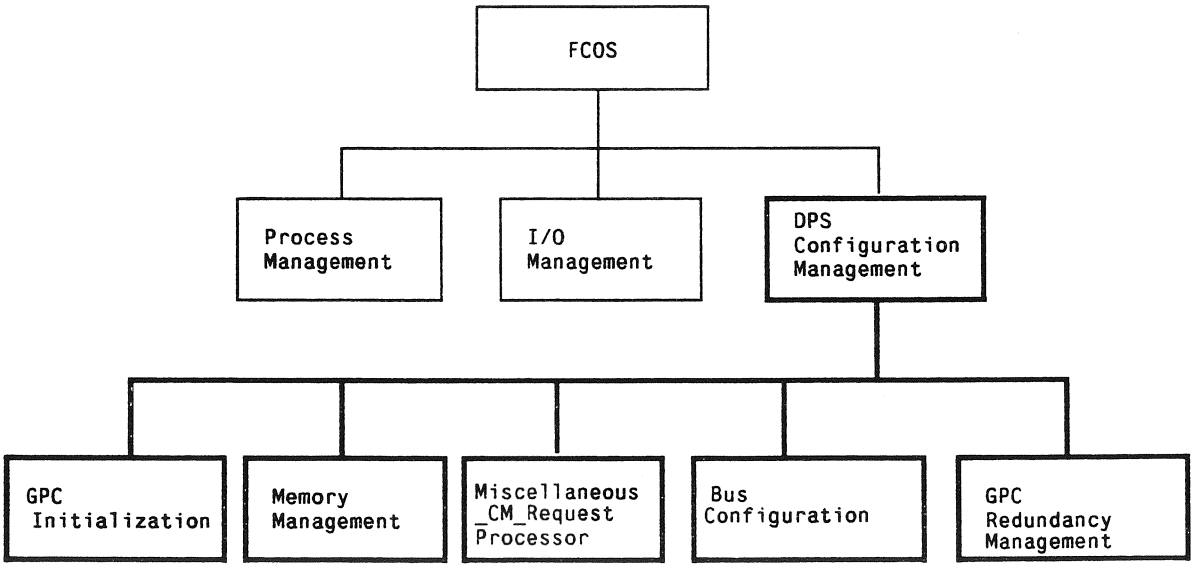


Figure 11.1-5.- DPS Configuration Management Hierarchy Diagram.

1. GPC Initialization

GPC initialization is the orderly sequencing of a GPC into an operational state having an acceptable confidence level. There are two types of GPC initialization that FCOS takes part in. The first type of initialization is initial program load (IPL) initialization that occurs when a user applies power to a GPC in HALT mode, performs the hardware IPL, then places the GPC in STBY mode.

The hardware IPL automatically loads the Bootstrap program from MM into a GPC. When the GPC mode switch is taken from HALT to STBY, the Bootstrap program loads the GPCIPL program. The GPC_IPL_Program (GPCIPL) which consists of the Self_Test_Program (STP) and the software system loader program (SSL). The STP performs GPC and IOP validity checks, and if no terminal errors are detected, passes control of the GPC to the SSL along with a system load identifier. The SSL uses the system load identifier to load a software system phase table into main memory from MMU. The SSL then uses the phase table to load into main memory the resident software and data areas of a primary avionics software system (PASS). After the resident software and data has been loaded, the SSL gives or relinquishes control of the GPC to the PASS or BFS. If control is given to the BFS, FCOS involvement terminates. If control is given to the PASS, the PASS resident software completes the IPL initialization. The PASS application software and data is then loaded into the GPC by the FCOS program overlay function upon request of an operation sequence (OPS) transition.

The second type of GPC initialization. Normal Initialization, is performed whenever a GPC has been previously IPL'ed. A normal initialization takes place when a user applies power to a GPC in the HALT mode, then places the GPC in STBY mode. The HALT to STBY mode switch causes the system reset PSW to be loaded giving control of the GPC to FCOS normal initialization processing. The System_Reset PSW is initialized with the FCOS normal initialization processing address as part of IPL initialization. Note, whenever power is applied to a GPC in RUN or STBY mode, the GPC will enter a WAIT state with interrupts disabled. The following is a list of GPC Initialization processes:

a. GPC_IPL_Program - A composition of the user supplied STP and the SSL. This program composition performs GPC and IOP validity check and loads the PASS resident software and data areas.

b. Software_System_Loader_Program - Performs the loading from MM of the PASS resident software and data areas. This program is a composition of the following FCOS programs.

   (1) Software_System_Loader - Controls the loading from MM of the PASS resident software.

(2) SSL_MM_MSC_Processor - Used by the Software_System_Loader to start the SSL_MM_BCE processor.

(3) SSL_MM_BCE_Processor - Used by the Software_System_Loader to load from MMU into the GPC the PASS resident software and data areas.

c. Normal_Initialization_Processor - Performs the GPC initialization of a previously IPL'ed GPC.

d. IOP_Initialization_Processor - Used during the IPL and normal initializations to initialize the OP.

e. PASS_GPC_Initialization_Processor - Initializes the PASS resident software and data areas.

f. FCOS_Run_Switch_Monitor - GPC STBY/RUN watchdog process.

g. OPS1_Normal_Initialization_Processor - Stub for regular initialization process in OPS1.

h. IPL_Bootstrap_Loader - Loads GPCIPL (STP and SSL) from the MMU into main memory.

2. Memory Management

Memory management maintains the proper main memory data/program configuration and provides for flight updates to the main memory, and provides the capability to disable (and re-enable) elements of BCE chains.

The GPC memory structure is discussed from a functional viewpoint to provide a basis for program overlays and program modifications. Program overlays are used to configure the main memory data/program organization. The source of program overlay is a mass memory unit or other GPC. Program modifications provide the capability to alter the contents of main memory units.

The FCOS and HAL/S compiler assume that all data and COMPOOL CSECTS (except HAL/S remote data CSECTS) reside in AP101 sector 0, which is always addressable, and if necessary one other data sector specified by the data sector register (DSR) field in the PSW. HAL/S remote data CSECTS can be located in any sector and are addressed via a special type address constant. A data or COMPOOL CSECT cannot span sectors. Code CSECTS can reside in any sector; however, only sector 0 and one other sector, specified by the branch sector register (BSR) field in the PSW, can be addressed directly. A code CSECT cannot span sectors.

Since the complete flight software package is too large to be permanently resident in main memory, the FCOS provides a program overlay capability to alter the data/program configuration during

flight. However, dynamic memory allocation; i.e., garbage collection, is not supported by the FCOS and all executable code and locations are determined preflight. Multiresidency of data and programs is allowed; however, these locations will be determined preflight, not computed, by the FCOS.

All FCOS control tables are allocated at system generation time using macros provided by FCOS. The use of macros provides the capability to parameterize the control table allocation procedure. The FCOS control tables are split between two CSECTS. The preferred storage area and communication vector table reside in a CSECT which is located at main memory address 0000. The remaining FCOS control tables reside in a control table CSECT that can be located in any data sector. The control table CSECT contains the following table groups:

a. Process control tables

b. Timer/event queue elements

c. I/O queue elements

Each group of tables occupy a contiguous area of memory. When these control tables are allocated, the individual control tables within each group are chained together to form free pool areas. These free pool areas contain all the unused control tables. As the various control tables are required by the FCOS, they are unchained from the appropriate free pool area and rechained in the appropriate active control table chain. When an active control table is no longer needed by the FCOS, it is unchained from its active chain and rechained in the appropriate free pool area for future use. The control table CSECT also contains the temporary stack areas and error log out areas. Memory management is divided into the following areas:

a. MM/GTG_Overlay_Processor - Provides the capability to change the configuration of main memory data and code.

b. MM/GTG_Overlay_BCE_Program_Generator - Builds the BCE program sequences required for a MM or GPC-to-GPC overlay request.

c. Program_Modification_Processor - Supports the additions to and modification of programs in main memory.

d. BCE_Element_Bypass_Processor - Processes requests to bypass or restore elements of BCE chains.

e. NSP_MTU_BCE_Element_Restore_Processor - Restores NSP and MTU bus control elements to their original states.

f. OPS_Uplink_Loader - Takes ground uplink data and writes it into core, a 16-bit halfword at a time. This process is used to facilitate an entry memory uplink (EMU).

3. Miscellaneous_CM_Request_Processor

The miscellaneous CM request processor is invoked to service requests for the reading or writing of certain discretes, the setting of the computer annunciation matrix (CAM) lights, or the setting/resetting of the termination control latches. When a configuration management request SVC is issued, the Miscellaneous CM_Request processor receives control from the SVC handling routine. It then determines which of the following three request types has been made:

a. The following types of requests (CM_SVC_Request=1) to read or write discretes will be accepted:

CM SVC operation	Operation description
1	Redundancy management of discrete register A
2	The RM Status_Register will be read.
3	Discretes will be set according to the mask input by the program making the request.
4	Discretes will be reset according to the mask input by the program making the request.

b. Two types of requests (CM_SVC Request=2) for CAM status light changes will be accepted:

CM SVC operation	Operation description
1	The fail vote lights can be turned off or on by setting and resetting the fail vote discretes. The mask passed to the program will be used to determine which lights should be turned on. Then the MSC_Set_Fail_Discretes_Program will be started to change the status of the lights.
2	The GPCs fail light can be turned on by loading the test register to simulate fail votes by other GPCs. The mask passed to the program will be used to determine which test fail votes to set.

c. For a CM SVC_Request of 3, restart sync trace logging.

4. Bus Configuration

The FCOS provides a set of services for the configuration of the 24 data buses which connect the input/output processor (IOP) with the interface units (IUs) via interface unit addresses (IUAs). The control logic for invoking these services resides in the DPS reconfiguration portion of system control. Configuration of the data buses is defined to be controlling the states (enabled or disabled) of the receivers and transmitters associated with each of the bus control elements (BCE) contained within the IOP.

The configurations of the receiver and transmitter associated with each BCE are:

Receiver enabled	Transmitter enabled	Mode
Yes	Yes	Command
Yes	No	Listen
No	No	Disabled

Generally, bus configuration is only concerned with enabling and disabling transmitters (changing a BCE status from command to listen mode and vice versa). However, facilities are provided to allow configuring a data bus to the disabled mode, or enabling a receiver (configuring a data bus from disabled mode to listen mode).

The FCOS also provides a set of services for the configuration of buses associated with logical strings. These requests are invoked by the system control redundant set bus configuration change which issues the appropriate service request(s) to FCOS.

Bus configuration provides bus mask management associated with bus/string configurations. These services may result from either direct request from applications/system control or from requests implied in other bus configuration requests.

Bus configuration includes processing for bus masking and GPC prime switching (if necessary) resulting from GPC forced-fail-to-sync and fail-to-sync. Bus configuration provides prime processing. The prime GPC is the GPC that downlists to toggle buffer 1.

FCOS maintains the current configuration tables which show transmitter and receiver status for each GPC. There are two types of current configuration tables. One shows the current status that should exist based on reconfiguration requests which have been serviced (requested current configuration table). The other indicates the actual hardware status of the transmitters and receivers (true current configuration table). Other bus configuration tables include the string definition tables indicating

the buses associated with each string, and the string mode indicators showing the current moding of each string. In addition to bus configuration tables, various GPC status information and masks are used.

There also exists a first-in/first-out bus configuration queue. The bus reconfiguration queue element (BRQE) contains the information from the parameter list which is needed to service the request. Nominally, this queue consists of zero or one queue element. It exists primarily as a place to store parameter list information when it is necessary to wait for I/O to become quiescent on the buses being reconfigured. However, it also serves to queue bus configuration requests when a request is made before a previous request(s) has been serviced. Bus configuration is divided into three areas:

a. Bus_Reconfiguration_Pre_Processor - Performs necessary preprocessing for bus reconfiguration requests.

b. Bus_Reconfiguration_Processor - Services bus reconfiguration requests.

c. Bus_Mask_Management - The setting/resetting of bus masks.

Note that there is no bus configuration initialization processor. The only bus configuration performed automatically by FCOS at initialization time is the enabling of transmitters and receivers for the mass memory buses. All other bus configuration needed at initialization time is performed as a result of bus configuration requests from system control.

5. GPC Redundancy Management

GPC redundancy management consists of GPC synchronization for both the common set and redundant set and fault detection identification. GPC synchronization is designed to keep the GPC's together with respect to time, data gathering, and internal queue manipulation. Because of possible dissimilar processing, the common set is synchronized with respect to time only. This is accomplished by four basic synchronization programs and one special program for SSIP (system software interface processor) sync initialization.

The special sync program, initial SSIP_Synchronization processor, is used when adding a GPC to the common set as part of the initial SSIP processing. It determines if any other GPC's are active and, if so, which ones are members of the common set.

The four basic sync programs are: Normal_SSIP_Synchronization_Processor (common set), I/O_Synchronization_Processor (common and redundant set), SVC_Synchronization_Processor (redundant set), and Timer_Synchronization_Processor (redundant set).

The Normal_SSIP_Synchronization_Processor is used to keep the common set synchronized via the minor cycle SSIP points. Its processing is initiated by every fourth SSIP timer interrupt. It also allows eligible GPC's to join the common set.

The I/O_Synchronization_Processor synchronizes I/O completion interrupts in the redundant set and provides input problem report (IPR) information for protected transaction processing.

The SVC_Synchronization_Processor is used by all SVC's in the redundant set that cause queue modification, data gathering, event modification, changing the state of a process, locked resources, data output, and time processing.

The Timer_Synchronization_Processor is used to synchronize the redundant set GPCs at every interval timer interrupt (PC2).

Synchronization communication between the GPCs is accomplished through the use of GPC sync discrete lines. Each GPC has 3 lines out (and hence 8 possible codes) and 12 lines in (3 from each of the other 4 GPCs). See figure 11.1-6 below for the code assignments.

Number	Binary	Meaning
0	000	Halt/Off/In-Sync for all routines
1	001	I/O complete with no error (IOC)
2	010	I/O complete with error (IPR)
3	011	***SPARE***
4	100	SVC interrupt
5	101	Timer interrupt
6	110	SSIP interrupt
7	111	Null/Run

Figure 11.1-6.- Synchronization codes.

To prevent a synchronization failure caused by GPC's arriving at different sync points at the same time (e.g., an SVC sync in one GPC and a timer interrupt in another GPC), a synchronization priority scheme is used. The only sync interrupt interference possible is between SVC, I/O complete, and timer interrupts. SVC synchronization is lowest priority and can be interrupted by both I/O complete and timer interrupts. I/O complete synchronization may be interrupted by timer interrupts only, while timer sync is not interruptible. Since normal SSIP sync is initiated by a timer interrupt, it is of the same priority as timer synchronization.

This allows sync processing in each GPC to migrate to the highest priority interrupt currently being processed by any GPC.

A sync history log is maintained for analysis of GPC synchronization failures. The SVC and I/O sync history entries are generated by the SVC_Synchronization_Processor and I/O_Synchronization_Processor, respectively. The timer sync entries are generated by the TQE_Expiration_Processor. The sync history log is "frozen" (i.e., further entries suppressed) at any fail-to-sync, except when that failure is caused by a GPC being manually placed in HALT, STBY, or power OFF. The sync history log can be restarted manually by a SPEC 0 ITEM 48 entry.

The key item in GPC discrete redundancy management (RM) is that the common set (CS) majority rules. The majority is just what it implies, i.e., greater than one-half of the GPC's identified as being in the common set. Therefore, the common set majority is defined as follows:

# GPCs in CS	CS Majority
5	3
4	3
3	2
2	2
1	1

The GPC discretes that are common set redundancy managed are:

Mass Memory 1 IPL Select (Disc Wd A)
Mass Memory 2 IPL Select (Disc Wd A)
Mass Memory 1 Ready (Disc Wd A)
Mass Memory 2 Ready (Disc Wd A)
BFS Run discrete (Disc Wd A)
BFS Engage Discretes for Disengage (Disc Wd B)
BFS BFC CRT Select A (Disc Wd B)
BFS BFC CRT Select B (Disc Wd B)
Sync Discretes for "000" Code (Not Downlisted)

Note that the I/O Terminate B discrete is not included in the above list.

Each time the discretes are to be RM'd, the number of GPCs in the CS is checked. If there is only one GPC in the common set, then the raw discretes become the redundancy managed discretes. If the CS contains two or more GPCs, then a count is taken of the number of GPC's detecting a change in a particular raw discrete. If the count of GPC's detecting a change is greater than the common-set-majority calculated for this case, then the changed discrete is entered as the RM'd discrete.

For example, if there are two GPCs in the CS, the CS-majority is defined as two. If one CS GPC "sees" MM1 go ready, but the other CS GPC still sees MM1 as busy, then the RM'd discrete would stay busy because the CS-majority of GPCs (i.e., both of them) has not seen MM1 go ready. Therefore, MM1 would not be available to either CS GPC in this case until both GPCs detected MM1 as ready.

Fault detection identification is accomplished within the redundant set by transferring sum words over the ICC buses and comparing them. GPC fail discretes are set when a GPC determines that it has had a series of consecutive miscompares with another GPC's sumword. GPC redundancy management is divided into the following areas:

a. Initial_SSIP_Synchronization_Processor - Provides synchronization when joining a common set.

b. Normal_SSIP_Synchronization_Processor - Synchronizes the common set at every SSIP point.

c. I/O_Synchronization_Processor - Synchronizes GPCs in a common or redundant set upon I/O completion interrupt.

d. SVC_Synchronization_Processor - Synchronizes GPCs in a redundant set for certain supervisor calls.

e. Timer_Synchronization_Processor - Synchronizes GPCs in the redundant set at every interval timer interrupt.

f. Sync_Fail_Processor - Performs necessary processing associated with GPC fail-to-sync or force-fail-to-sync conditions.

g. Sync_Fail_CAM_MSC_Processor - MSC processor to light CAM lights when a fail to sync is received.

h. Fault_Detection_Identification - Compares sum words for all GPC's in the redundant set.

i. Fail_Discretes_MSC_Processor - MSC processor used to light CAM lights when failure is noted by Fault_Detection_Identification.

j. GPC_Discretes_Redundancy_Manager - Compares discretes from all GPCs in the common set and provides one discrete word for all to use.

k. Sync_Mask_Build_Routine - A service routine used by each of the synchronization programs mentioned above.

l. Sync_Fail_Interface_Routine - Builds parameter list need by the sync fail processor.

m. Sync_History_Log_Routine - A service routine that logs sync history entries in the sync history log.

E. References

1. Space Shuttle Programs Orbiter Avionics Software Operational (DBS) Detailed Design Specification (DDS) Volume: II - System Services, Part: 1 - Flight Computer Operating System (FCDS).

11.1.1 TIME MANAGEMENT[1]

Time Management includes the FCOS support for all HAL/S real-time statements which have time as a parameter, the FCOS support of master timing unit (MTU) updates, MTU/software clock synchronization, the FCOS support of a software backup to the MTU, and the FCOS support of timer initiated I/O.

In support of these functions, the FCOS builds table entries called Timer Queue Elements (TQEs). A TQE contains information which identifies its purpose, the time at which it is to expire, and the process to which it applies. When a TQE has been initialized with this information, it is said to be active. Active TQEs are chained together and arranged by time such that the top TQE in the chain is next to expire. Active TQEs contain the time at which they are to expire.

Program Counter 2 (PC2) is used as an interval timer to cause interrupts at the times specified in the active TQEs. PC2 is always set using the top TQE in the active TQE chain. When PC2 expires, the information in the top TQE is used to identify what action is to be taken.

FCOS maintains a Greenwich Mean Time (GMT) clock as a software backup to the MTU. The software GMT clock is also used to provide finer granularity and faster response in the processing of HAL/S schedule and wait statements which have time options. The software clock has microsecond granularity and is maintained in the Communications Vector Table (CVT). The clock is updated using Program Counter 1 (PC1) and the Systems Software Interface Processing (SSIP) TQE interrupt (every 24th cycle).

The SSIP TQE interrupt is also used to initiate updates to the software clock and to initiate the FCOS duty cycle computation. Since the software clock is only updated periodically, current GMT or MET is determined by summing the software clock value and the difference between the maximum PC1 value and the current PC1 value. The FCOS duty cycle computation is also performed every 24th SSIP TQE interrupt.

Whenever a request to perform an action at a time it is made, a TQE is built and passed to the Timer Queue Generator process to add the TQE to the active TQE chain.

The Timer Dequeue Processor is entered whenever a process is removed from the process run queue.

Time Management is also entered whenever PC2 expires.

The functional description of Time Management presented in the following paragraphs is divided into the following areas (fig. 11.1.1-1):

- Timer Queue Generator - The process of creating the active TQE chain
- GPC Clock Expiration Processor - Results of a PC1 interrupt

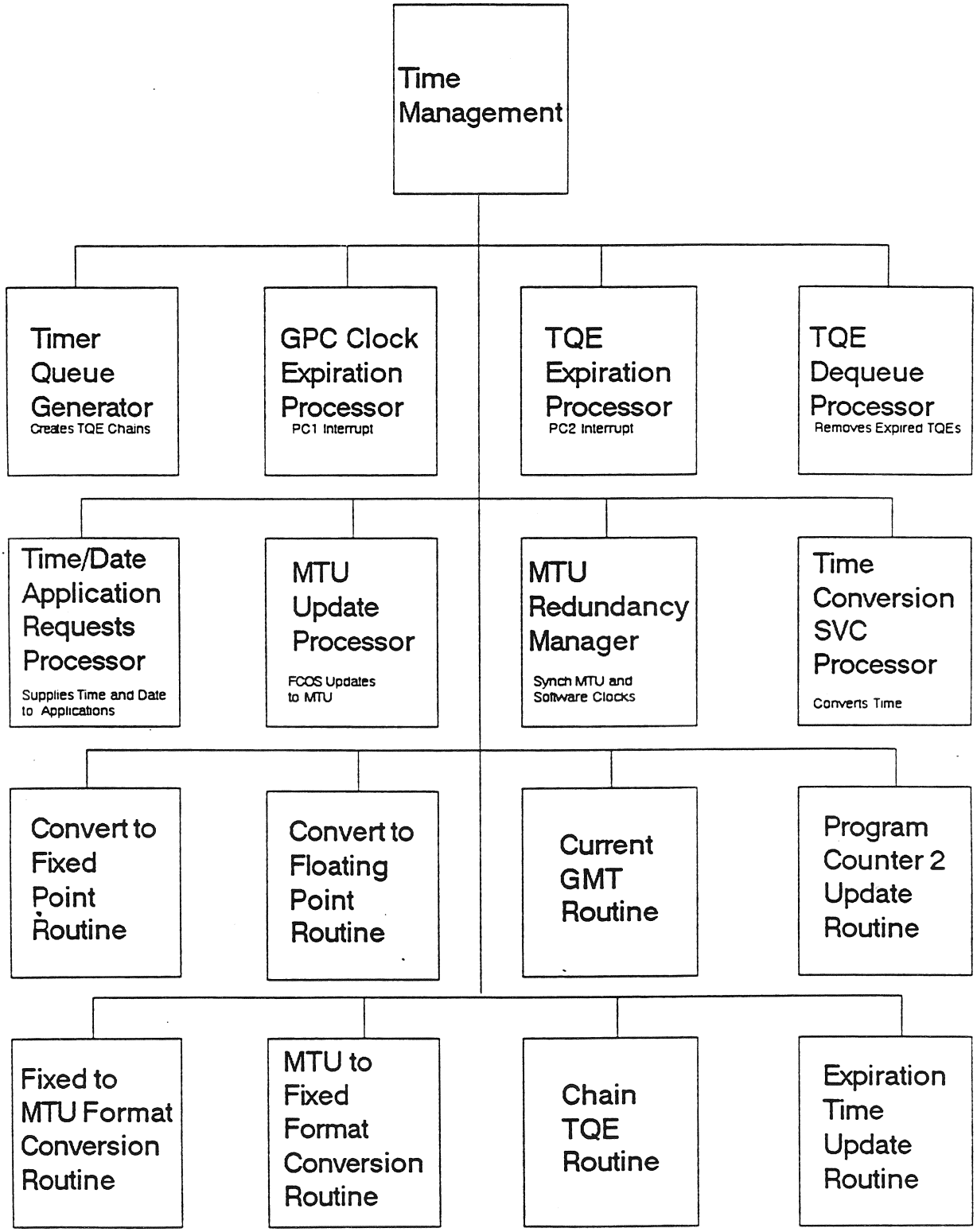


Figure 11.1.1-1.- Time Management Systems Software.

- TQE Expiration Processor - Results of a PC2 interrupt
- TQE Dequeue Processor - Removing expired TQEs from the active TQE chain
- Time/Date Application Requests Processor - Supplies time and date to requesting applications
- MTU Update Processor - FCOS involvement in updates to the MTU
- MTU Redundancy Manager - Synchronization of MTU and software clocks
- Time Conversion Supervisor Call (SVC) Services - Handles conversion SVCs
- Convert To Fixed Point Routine - Conversion routines
- Convert To Floating Point Routine - Conversion routines
- Current GMT Routine - Supplies current GMT
- Program Counter 2 Update Routine - Updates PC2
- Fixed-to-MTU Format Conversion Routine
- MTU To Fixed Format Conversion Routine
- Chain TQE Routine - Handles TQEs
- Expiration Time Update Routine - Updates expiration time

A. Timer Queue Generator

The Timer Queue Generator places interval timer requests on the time queue.

This routine figures out where the new TQEs will be loaded and executes the appropriate subroutine based on the TQE Type Indicator that is set up by the calling program.

For TQE Type Indicators of 1 and 8, CTOX For Repeat Every is performed to compute TQE Time of Expiration for a Repeat Every TQE. If the Cycle Overrun Indicator is not ON, then the Time Initiated I/O Parameter List is Checked for timer initiated I/O and the I/O SVC Service Processor or Pre-initialized I/O SVC Processor is called if there is timer initiated I/O. The Repeat Delta Time is obtained from the Process Control Table (PCT) and Expiration Time Update Routine is called to update the TQE Time of Expiration with the Repeat Delta Time.

For TQE Type Indicators of 2 and 5, CTOX for Wait Schedule In is performed to compute TQE Time of Expiration for a Wait or Schedule in TQE by adding the input delta time to the time from the Get Top TQE Time Routine.

For TQE Type Indicators of 3 and 4, CTOX for Wait Until Schedule is performed to compute TQE Time of Expiration for a Wait Until or a Schedule At TQE. The time is obtained and converted to fixed point format by the Convert To Fixed Point Routine and saved in TQE Time of Expiration.

For TQE Type Indicator 6, CTOX for Schedule Until is performed to compute TQE Time of Expiration for a Schedule Until TQE. The time is converted by Convert To Fixed Point Routine and saved in TQE Time of Expiration.

For TQE Type Indicator 9, Queue Mass Memory TQE is performed. Current time is obtained by Get Top TQE Time, updated by the Expiration Time Update Routine and saved in TQE Time of Expiration.

For TQE Type Indicators of 10 and 12, CTOX for Runtime GMT Requests is performed to compute TQE Time of Expiration for a Runtime or GMT update request. The coincident time is passed in the TQE Parent PCT Address. The coincident time is adjusted to the proper format and saved in TQE Time of Expiration. Update Time is called to add in the Software Clock Offset to the TQE Time of Expiration. The TQE Time of Expiration is adjusted so that this TQE will expire under an SSIP timer interrupt.

For TQE Type Indicator 11, CTOX for MET Update is performed to compute the TQE Time of Expiration of a MET update request. The MTU Update Time is obtained and saved in TQE Time of Expiration. The TQE Time of Expiration is adjusted so that this TQE will expire under an SSIP timer interrupt.

For TQE Type Indicator 13, CTOX for I/O Delay is performed. Current time is obtained by Get Top TQE Time and added to the I/O delay time in the TQE Time of Expiration by the Expiration Time Update Routine.

After TQE Time of Expiration has been computed for the appropriate TQE Type Indicator, the TQE Request Type is checked to see if this is an initial TQE. If this is not an initial TQE, the Chain TQE Routine is called to requeue the TQE to the chain. Otherwise, current time is obtained by Get Top TQE time. Next, the TQE Time of Expiration is compared to the current time. If the TQE Time of Expiration is less than or equal to the current time, then the TQE is considered past due. If the TQE is not past due, Add TQE is called to add the TQE to the active TQE chain. If the TQE is a Schedule At and Repeat Every, then it is Phase Scheduled.

Phase Scheduling is the computation of a new Schedule At time which would be the next repeat interval in the future for the process. Compute New At Time is called to handle this.

Compute New At Time computes the new at time by figuring the number of intervals between the old at time and the current time and adds one repeat interval. If this time is in the future and the process is not SSIP, one additional repeat interval is added.

B. GPC Clock Expiration Processor

The GPC Clock Expiration Processor handles PC1 interrupts and reloads the PC1 clock.

C. TQE Expiration Processor

The TQE Expiration Processor is responsible for fielding GPC PC2 interrupts and initiating the required action.

D. TQE Dequeue Processor

The TQE Dequeue Processor scans the active TQE chain for any TQEs associated with the PCT being cancelled or terminated.

E. Time/Date Application Request Processor

The Time/Date Application Request Processor services requests for runtime, clocktime, date, and MET.

F. MTU Update Processor

The MTU Update Processor supports user-directed updates to the MTU and allows all time sources to be set equal to the source currently selected by the prime GPC.

The MTU Update Parameter List parameters are validity-checked for possible ICC pollution. If an error is detected, an error is logged and the update is ignored.

For MTU Update Request Type 1, the MTU Update Processor will perform a MET Reset. MET Reset initializes the I/O Buffer Address, Operation Code and the Event Address in the I/O SVC Parameter List. Then MET Reset calls the I/O SVC Service Processor to issue the I/O request. On return from the I/O SVC Service Processor, the Event Address is zeroed.

For MTU Update Request Type 2, the MTU Update Processor will perform a MET Update. MET Update initializes the I/O Buffer Address and Operation Code in the I/O SVC Parameter List. Then Convert to Binary Coded Decimal (BCD)is executed to convert the MTU Update Coincident Time from minutes to BCD format. The Current GMT Routine is called to get current GMT time. Current GMT time is converted to MET time and the MET MTU Update Coincident Time is computed. This time is saved in the TQE at the TQE Free Pool Address. Then Convert Delta is executed to add MET MTU Coincident Time to MET MTU Update Delta Time. Flags are set in the TQE at the TQE Free Pool Address to indicate a MET update. Next, Common MET/GMT Update Logic is executed to perform the common processing for a MET/GMT update.

The Common MET/GMT Update Logic gets the MTU Update Absolute Time and calls the Convert to Fixed Point Routine to convert MTU Update Absolute Time to FCOS fixed point format. MTU Update Absolute Time is then saved

in MTU Update Time. The MTU Update Absolute Time is then converted to MTU format by calling the Fixed to MTU Format Conversion Routine. Next the I/O SVC Service Processor is called to process the I/O request.

On return from the I/O SVC Service Processor, a TQE is obtained from the TQE free pool at the TQE Free Pool Address. The MTU Update Coincident Time is stored into TQE Parent PCT Address of the TQE. Finally, Timer Queue Generator is called to enqueue the TQE.

For MTU Update Request Types 3 and 4, the MTU Update Processor will perform a GMT Update. GMT Update initializes the I/O Buffer Address and Operation Code in the I/O SVC Parameter List. The MTU Update Coincident Time is then converted to BCD format by calling Convert to BCD. Then MTU Delta Update Time is checked to see if it is greater than 15 milliseconds. If the update is larger than 15 milliseconds, the TQE is flagged as an initial GMT update, otherwise it is flagged as a Runtime update. Convert Delta Time is executed to convert MTU Delta Update Time to FCOS time format. Then the Common MET/GMT Update Logic is executed.

For MTU Update Request Type 5, the MTU Update Processor executes Accumulator Sync. Accumulator Sync starts by initializing the I/O Buffer Address in the I/O SVC Parameter List and gets the selected time source from the ICC buffer of the requesting GPC.

The I/O Buffer Address and Operation Code are initialized in the I/O SVC Parameter List. The I/O SVC Service Processor is called to execute the I/O request to reset the MTU accumulators. On return from I/O SVC Service Processor, a TQE is obtained from the TQE Pool Address and the Current GMT Routine is called to get the current time. The current time is stored into the TQE Parent PCT Address field and the Update Time is executed to add in the Software Clock offset to the TQE Time of Expiration. Update Time is executed again to add one additional minute to the TQE Time of Expiration. Finally, the TQE Flags are initialized with the accumulator sync TQE Type Indicator.

After handling the selected time source, Accumulator Sync continues by initializing the I/O Buffer Address and Operation Code again. The Event Address is also zeroed. Next, the TQE Time of Expiration is converted to MTU format by calling the Fixed To MTU Format Conversion Routine. Then the I/O SVC Service Processor is called to process the I/O request followed by a call to Timer Queue Generator to enqueue the TQE.

If no bus was available, accumulator 1 is assumed. The I/O requests will be issued by the MTU Update Processor but no I/O will be started by I/O management.

G. MTU Redundancy Manager

The MTU Redundancy Manager is responsible for keeping each GPC internal clock synchronized with the MTU or with the prime GPC's internal time if the MTU is unavailable or out of tolerance. Each GPC executes MTU RM independently and may arrive at a different time source than other

common set GPCs. If this occurs, an FTS may occur in the near future. Redundancy Manager also supports internal GMT (RUNTIME) initialization, and timetagging of applications data. The MTUs are read on SSIP Cycle 15. This data is moved to the ICC buffer in SSIP cycle 16. This is ICC'd to all common set GPCs on SSIP Cycle 18. On SSIP Cycle 19, MTU RM is initiated.

There are two entry points for the MTU Redundancy Manager: SVC Sync Routine and an alternate entry point.

For MTU RM Request Type 1, the first GPC is initialized by Initialize First GPC. The I/O Terminate Discrete in the Discrete Input A word is checked to see if the BFS is engaged.

If the BFS is engaged, buses to the MTU are not available for use; therefore, time is initialized from the PCMMU. A BCE program is started which loops reading the PCMMU which is updated from the MTU at a five times per second rate. While the PCMMU is being read, the GPC loops looking for a new time from the PCMMU. When the new time is received, it is limit-checked by the MTU Limit Check Routine to see if the time is reasonable. If the PCMMU time passes the limit check, then the GPC is initialized to that time.

If buses to the MTU are available, then time is initialized from the MTU. There are three MTU accumulators. Each accumulator is checked until a good one is found or all have been checked. Each MTU accumulator is checked to see if it is comfaulted; the accumulator is then limit-checked. If a valid MTU accumulator is found, GPC time will be initialized to that MTU accumulator.

If no valid time source is located, then GPC time will be initialized to GMT Day 1 and MET of 0. The new time is saved in three places: GPC Software Clock, MET Reference Time, and Prime GPC Time.

For MTU RM Request Type 2, a secondary GPC is initialized by performing Initialize Secondary GPC. Time is initialized in secondary GPCs from the ICC buffer of the lowest GPC other than self. The time and Time Source is obtained from the lowest GPC. Other fields which are initialized are MET Reference Time and Clocktime.

For MTU RM Request type 3, Time Management Processing is performed. Figure 11.1.1-2 illustrates Time Management Processing that is described in the paragraphs below. The MTU BITE status is read and prepared for downlisting. The GPC then gets the selected Time Source from its ICC buffer and then checks to see if a different time source has been selected.

If the selected Time Source is an MTU accumulator, the Find Valid MTU routine is performed. Find Valid MTU loops through the ICC buffers in the common set looking at the transmitter status for the GPC with the selected accumulator. When the ICC buffer is found, the COMFAULT data

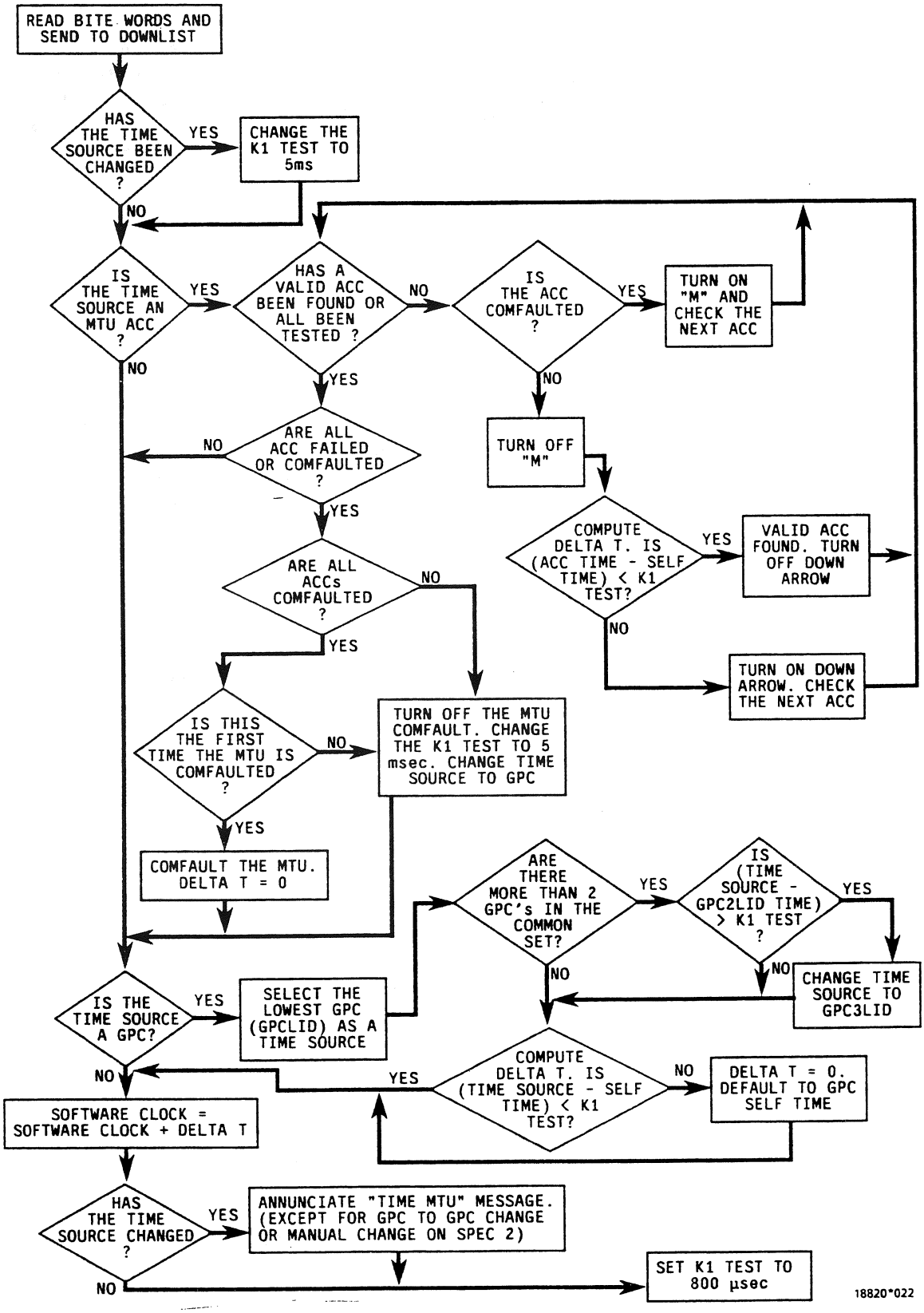


Figure 11.1.1-2.- Time Management Processing.

is checked. If the MTU COMFAULT flag is on, the "Data Missing" flag is turned on; if not, it is turned off. The MTU data is then differenced with the Timetag. This difference is added to the difference between the GPC (with the MTU data) and self GPC time at the exit from the common set sync. If this difference is greater than the K1 Test (800 microseconds), then the down arrow is turned on and the search is restarted for the next accumulator. If the difference is less than the K1 Test, then the down arrow is turned off and this accumulator is accepted as good. This process starts with the selected accumulator; if the selected accumulator fails the test, then the next accumulator is checked. A GPC time source is not attempted until all three accumulators have failed. An accumulator that has failed in the past and is now available is considered a valid candidate for a time source. When all MTU accumulators have failed, the K1 Test is opened to 5 milliseconds when trying to select a GPC Time Source.

If the Time Source is a GPC, then the number of GPCs in the Common set is obtained. If there are less than three GPCs, the time of GPCLID (the lowest numbered GPC) is differenced with self GPC time. If the difference is greater than the K1 Test (5 milliseconds), then self GPC time is the Time Source, otherwise the Time Source is GPCLID.

When there are three or more GPCs, a "Try" time source is selected. First the two lowest GPC times are differenced and compared to the K1 Test (5 milliseconds). If the difference is less than or equal to the K1 Test, then the "Try" time source is GPCLID; otherwise it uses GPC3LID as the "Try" time source. Once the "Try" time source has been selected, its time is differenced with self GPC time. If the difference is less than or equal to the K1 Test, then the "Try" GPC is the selected Time Source; otherwise, the self GPC time will be the Time Source.

If a GPC is using a Time Source other than itself, such as the MTU or another GPC, then the difference between the GPC and its Time Source is saved in the Software Clock Offset. The Software Clock Offset is added to the GPC Software Clock each pass through Time Management Processing to adjust for drift between the GPC time and the selected Time Source.

If a new time source has been selected, it is saved and a "TIME MTU" message is annunciated to show that the selected Time Source has changed and the K1 Test is dropped back down to 800 microseconds.

For MTU RM Request Type 4, Timetagging is performed. Time is obtained from the Last Expired TQE Time and the Convert To Floating Point Routine is called to convert time to floating point format. The time in floating point is then saved eight halfwords before the Address of I/O Buffer.

H. Time Conversion SVC Services

Time Conversion SVC Services is a time conversion routine which will convert time in MTU format or fixed point format to floating point

format and then stores the converted time at an address specified by the caller.

I. Convert To Fixed Point Routine

The Convert To Fixed Point Routine converts time from floating point to fixed point format.

J. Convert To Floating Point Routine

The Convert To Floating Point Routine converts time in fixed point format to time in floating point format.

K. Current GMT Routine

The Current GMT Routine computes the current MET.

L. Program Counter 2 Update Routine

The Program Counter 2 Update Routine computes timer intervals and updates Program Counter 2 (interval timer). Program Counter 2 will be updated with a value to expire at the time of expiration of the TOP TQE.

M. Fixed To MTU Format Conversion Routine

The Fixed To MTU Format Conversion Routine converts time in fixed point to MTU format.

N. MTU To Fixed Format Conversion Routine

The MTU To Fixed Format Conversion Routine converts time in MTU format to fixed point format used by FCOS.

O. Chain TQE Routine

The Chain TQE Routine inserts TQEs into the TQE chain in the proper position. This module will add a TQE to the TQE chain in the proper place. If the time of expiration of the New TQE is less than that of the first TQE in the chain, then the New TQE will be added as the first TQE in the chain. Otherwise, the Chain TQE Routine will loop through the TQE chain until it finds a TQE whose time of expiration is greater than that of the New TQE. The New TQE will then be removed from the free pool, added to the chain, and the TQE Free Pool Address will be updated.

P. Expiration Time Update Routine

The Expiration Time Update Routine updates the TQE Time of Expiration fields in the Timer Queue Element.

Q. References

1. System Software Design Specification, section 4.1

11.1.2 Error Log Reset

11.1.2.1 Overview

Computers are not perfect and can fail. There are two forms of computer failure. The first is called a "hard fail" in which the computer will not run, and the second is called a "transient failure" in which the computer fails but can be re-established by doing an IPL or some other action (in the lab perhaps by cycling power). Hard failures or transient failures must be explained if the box is ever to fly again. To aid in doing this, there are several resources available. IBM has a very sophisticated factory test program that can check virtually every circuit in the computer; however, to get a leg up on this, it is very important to look at the computer's own software state at the time of failure if possible. This is the purpose of the various error logs that are maintained in the GPC memory. There are four error logs maintained in memory:

A. Sync Trace History Log

B. I/O Error Log

C. GPC Error Log

D. Auxiliary GPC Log

Only the first three error logs are associated with the Error Log Reset and will therefore be discussed.

11.1.2.2 Purpose of Error Log Reset

The Error Log Reset is initiated by an Item 48 EXEC on the GPC MEMORY SPEC (see fig. 11.1.2-1). The execution of this item causes the following to happen:

A. Restart the Sync Trace History Log

B. Clear the I/O Error Log

C. Clear the GPC Error Log

D. Clears the Computer Annunciation Matrix (CAM) lights

The Sync Trace History Log maintains a record of events at every sync point. These events consist of the following:

A. Time of fail-to-sync (F-T-S).

B. Address at which the interrupt occurred.

C. What type of interrupt caused the F-T-S (i.e., Supervisor Call, I/O Complete, Timer Sync, I/O Problem Report, or Common Set Sync Points).

The Sync Trace History Log is frozen when a GPC detects a fail-to-sync except when this is due to the mode switch going to standby or halt, or a power-off.

The I/O Error Log maintains a record of events when the GPC detects an I/O Error. These events consist of the following:

A. Time of error

B. IOP status register

C. Word count where error occurred (also called the residual word count and gives the location in a transaction where the I/O ERROR occurred)

D. BCE number

E. BCE element

F. Device ID/OP code

The I/O Error Log is cleared when an Error Log Reset is done, this contrasts with the I/O RESET which reinitiates all the I/O traffic and does not clear the I/O Error Log. The last I/O Error is also stored in another area of memory and is not cleared. The reason for clearing the I/O Error Log is to reinitialize the first 5 slots of the 10-slot I/O Error Log. The first five slots store the first five I/O Error Log entries which occur after this initialization and are not overwritten when more errors occur. This contrasts with the last 5 slots which may be overwritten given enough errors (for example 11 errors after reinitialization, first 5 stored "permanently", second 5 temporarily, with the 11th error entry overwriting slot 6). Thus, if an I/O Error kicked off a large sequence of errors, you might have a chance of seeing the I/O Error that caused it all as it could have been stored in the "permanent" slots. Beginning with OI-21, the GPC Error Logs will be processed identically to the I/O Error Log and cleared upon the ITEM 48 execution. The 10-deep stack consists of permanent and temporary areas with the first 5 remaining static and the last 5 being a circular stack.

The CAM advises the crew on GPC failures (see fig. 11.1.2.-2). The lights are driven by fail discretes and the Error Log Reset turns the lights off by resetting these discretes. There is an implication to our recovery procedures because of this fact. There are two kinds of IPL, the so called menu IPL and the default IPL. If fail votes are set, the default IPL will fail but the menu IPL can be used as a workaround. Also if the GPC has fail votes against it, extra steps must be taken to make that GPC join the redundant set after the IPL. Therefore, to keep recovery procedures clean we usually do an Error Log Reset at the beginning of the procedure.

11.1.2.3 SOURCES

A. Level A CPDS, SS-P-0002-170J, 5/27/88.

B. DDS, NAS 9-16920, 9/9/87.

C. STS-29 Final Software Listings.

D. Conversations with IBM, (Marty Sommers).

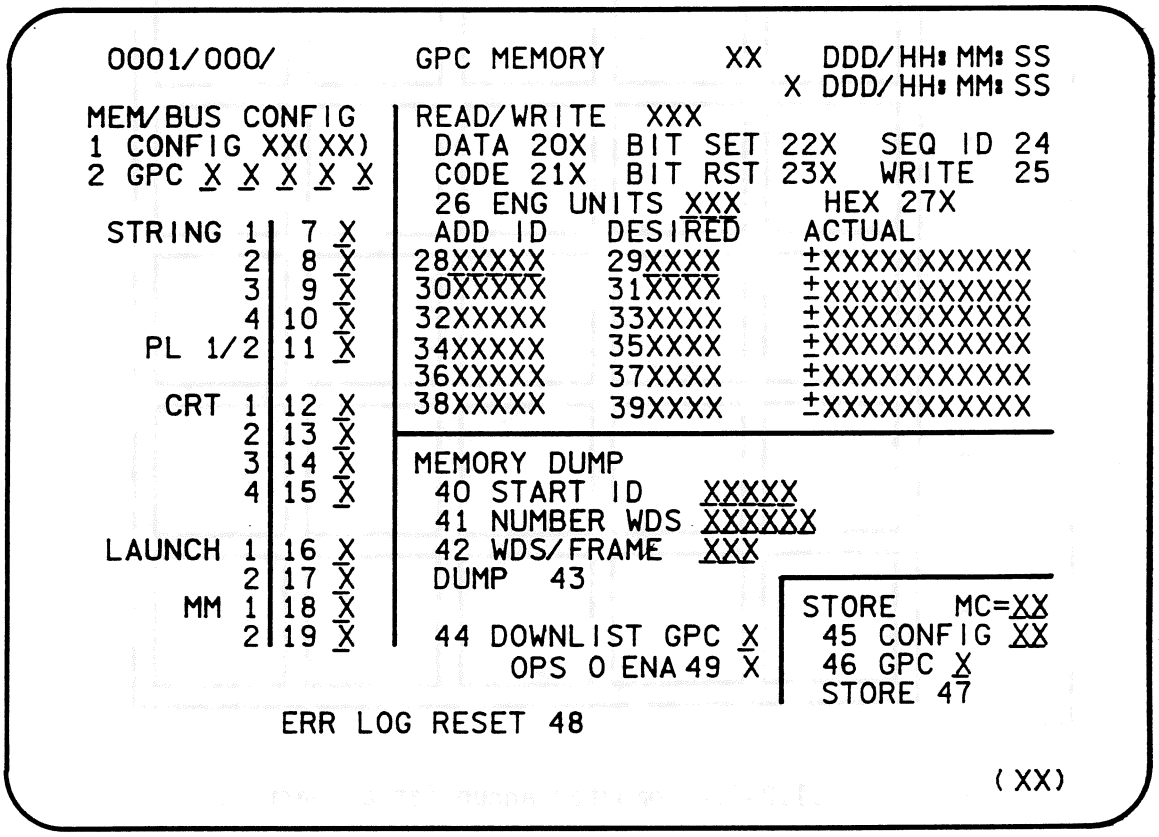


Figure 11.1.2-1.- GPC MEMORY SPEC.

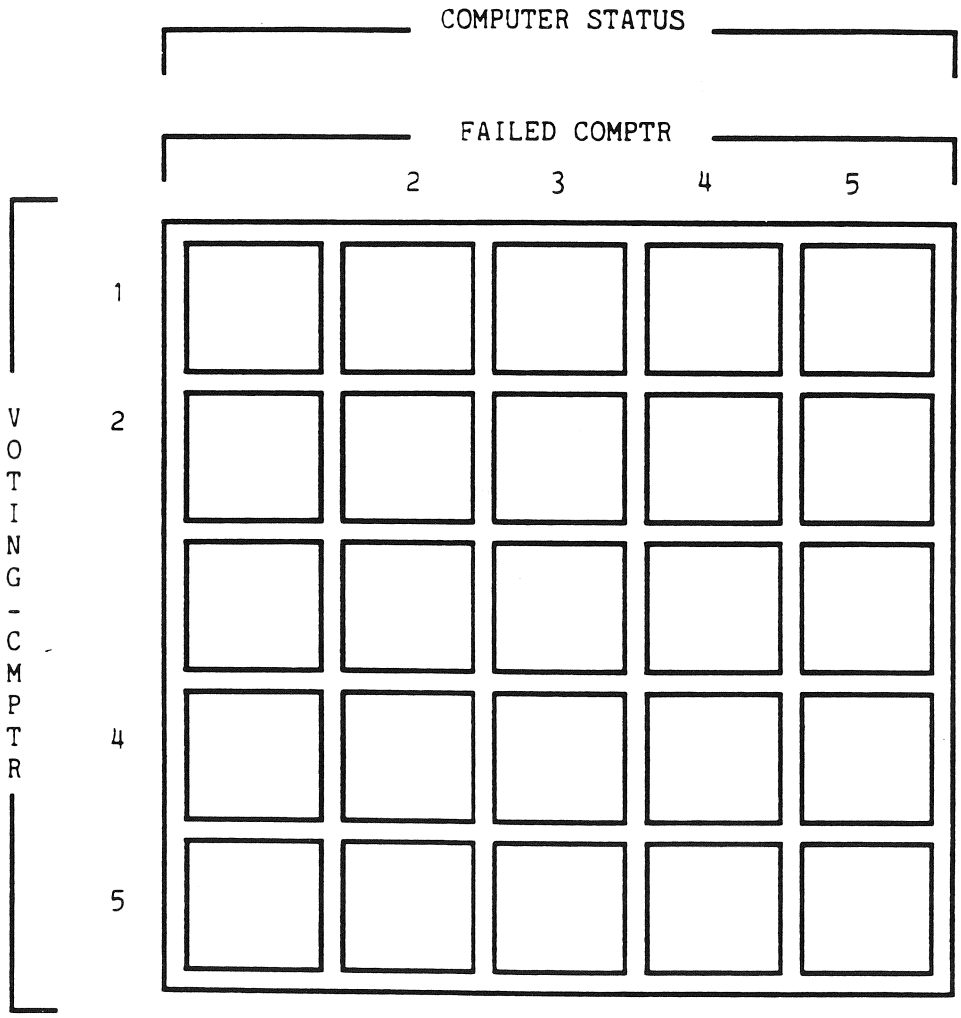


Figure 11.1.2-2.- Computer Annunciation Matrix.

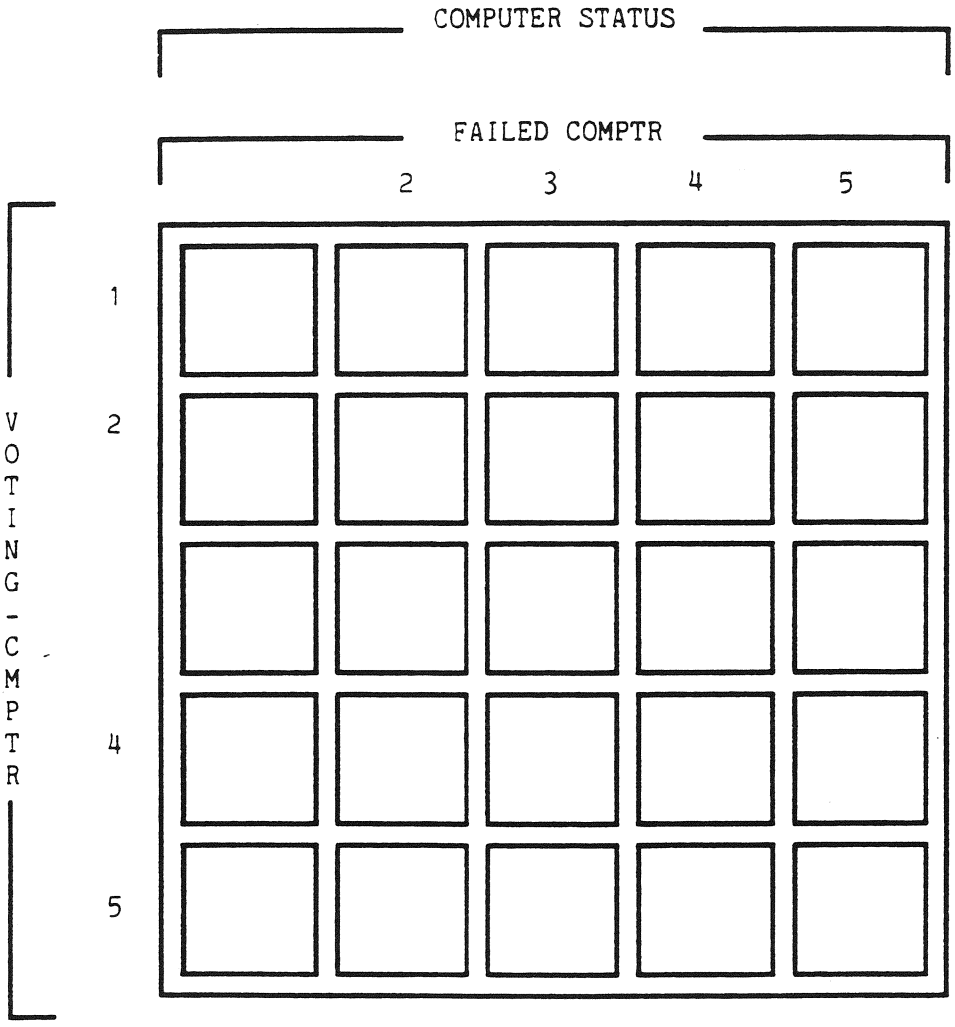


Figure 11.1.2-2.- Computer Annunication Matrix.

11.1.3 HISAM Dump Software

11.1.3.1 Overview

HISAM dump stands for hardware initiated stand alone memory dump. This feature potentially allows the memory of an ailing GPC to be dumped. The requirements state that a HISAM dump will begin when: the dump discrete is set; I/O term B is set; and the mode switch transitions from HALT to STBY. These requirements are to be implemented in any software loaded in the GPC.

To eliminate some potential confusion, the HISAM dump is not a hardware dump of GPC memory. It is a hardware initiated software dump of GPC memory.

The requirement for HISAM dump capability in any software loaded in the GPC means that there are three versions of HISAM dump software, one for each type of software that may be loaded in a GPC. The manner in which the HISAM dump requirement is implemented varies among the software types. The first type of software is the software (SELF TEST PROGRAM and SOFTWARE SYSTEM LOADER) that is loaded as a result of the hardware IPL. This software provides the capability to select and load the other two types of software: the PASS or BFS software. The hardware IPL and PASS HISAM dumps are the only HISAM dumps discussed here. In addition, the dump format will be discussed.

11.1.3.2 Hardware IPL HISAM Dump

When the mode switch is transitioned from HALT to STBY, control is passed to the self test program (STP) via the system reset program status word (PSW). The STP checks to see if the HISAM dump requirements are met: in STBY, I/O term B set, and dump switch bit set. If these conditions are met, the HISAM dump software (MSCHISAM)is executed. Figure 11.1.3-1 show the details of the STP software flow.

MSCHISAM is a program which is exectued by the MSC. It is written as an inifinite loop, dumping memory in a circular manner until the GPC is HALTed. Each iteration of the loop builds a 32 word frame to be downlisted. The format of the data in the frame is described in the Dump Format section below. Once the frame has been built and stored in the output buffer, the BCE program (BCEHISAM) is called. Figure 11.1.3-2 and table 11.1.3-I describe the details of the MSCHISAM software flow.

BCEHISAM is a BCE program which transmits a frame of data to the PCMMU, toggle buffer 5. Figure 11.1.3-3 and table 11.1.3-I describe the details of the BCEHISAM software flow.

NOTE: Memory location 0 is never dumped with this implementation of the HISAM dump. This may be fixed in a future version of the software.

11.1.3.3 PASS HISAM Dump

The PASS HISAM dump is functionally the same as the hardware-IPL HISAM dump with three exceptions:

A. The PASS HISAM is sent to toggle buffer 1 of the PCMMU, as opposed to toggle buffer 5 for a hardware-IPL or BFS HISAM dump.

B. The PASS HISAM dump frame length is equal to 112 words, as opposed to 32 words for a hardware-IPL or BFS HISAM dump.

C. The PASS dump format id is 93, as opposed to 90 for a hardware-IPL or BFS dump.

The dump is described in more detail below.

When the mode switch is transitioned from HALT to STBY, control is passed to the initialization routine (FCMINIT) via the system reset PSW. This routine checks to see if the HISAM dump requirements are met in STBY, I/O term B set, and dump switch bit set. If these conditions are met, the HISAM dump software is executed (FIOHISAM). Figure 11.1.3-4 shows the details of the FCMINIT software flow.

FIOHISAM is a program which is executed by the MSC. It is written as an infinite loop, dumping memory in a circular manner until the GPC is halted. Each iteration of the loop builds a frame of 112 words to be downlisted. The frame will contain header words and data words as described in the Dump Format section below. Once the frame has been built and stored in the output buffer, the BCE program FIOHSMPG is called. Figure 11.2.3-5 and table 11.1.3-II describe the details of the software flow.

FIOHSMPG is a BCE program which transmits a frame of data to the PCMMU, toggle buffer 1. The contents of the output buffer are transmitted 32 (or less) words at a time until all 112 words have been transmitted. Figure 11.1.3-6 and table 11.1.3-II describe the details of the software flow.

Note: The initialization routine (FCMNINIT) is also executed when the mode switch transitions from RUN to STBY. This will result in a HISAM dump being performed if all other requirements are met. This is contrary to the requirements and will be fixed in a future version of the flight software. The BFS follows requirements and will only start a HISAM dump on a HALT-to-STBY mode switch transition if all the other requirements are met.

11.1.3.4 Dump Formats

A. Dump rates

The Orbiter downlist can be transmitted at one of two rates. High data rate (HDR) is at 128 kilobits per seconds (kbps), and low data rate (LDR) is at 64 kilobits per second (kbps). Each data rate is divided up among the various downlists - GNC, SM, PL9, BFS, OI, and PAYLOAD. Thesum of the kbps for the individual downlists will equal the total kbps for that data rate.

The kbps are directly related to the words per frame based on the GPC output rate. The GPC puts out data at a rate of 25 GPC frames per second. The following equation specifies the relationship between kbps and words per frame:

$$\text{words/frame} = \frac{\text{kbps}}{25 \text{ GPC frames/sec} \times 16 \text{ bits/word} \times 1 \text{ kilobit/1000 bits}}$$

$$\text{words/frame} = \frac{\text{kbps}}{0.4}$$

This relationship is used to determine if the current telemetry format load (TFL) can support a dump at a specific words-per-frame rate.

The PASS GPC HISAM dump will always contain 112 16-bit words per frame. According to the above equation, 112 words per frame requires a bandwidth of at least 44.8 kbps. Since the HISAM dump is downlisted through toggle buffer 1, the TFL must have at least 44.8 kbps allotted to the GNC downlist.

The hardware IPL HISAM dump will always contain 32 16-bit words per frame. According to the above equation, 32 words per frame requires a bandwidth of at least 12.8 kbps. Since this dump goes through toggle buffer 5, the TFL must have at least 12.8 kbps allotted to the BFS downlist.

B. Dump contents

In frames 0 and 25, words 1 through 6 of the frame consist of the data cycle header which contains the following information:

Word	Description
1	Sync pattern
2	Frame counter and format id
3	Vehicle id, GPC id, and mission id
4,5,6	GMT

In frames 1 to 24 and 26 to 49, words 1 and 2 of the frame consist of the frame header which contains the following information:

Word	Description
1	Sync pattern
2	Frame counter and format id

In addition, the following header words are added to the data cycle and frame header:

Data cycle word	Frame header word	Description
7	3	Dump frame length in 16-bit words per frame
8	4	Number of words in the frame (not including checksum word)
9,10	5,6	Memory address of first memory dump word in the frame
11,12	7,8	Memory address of last memory dump word in the total dump

The remaining words in the frame contain main memory contents, except the last word in the frame contains the checksum of the entire frame.

Figure 11.1.3-7 summarizes the formats of the HISAM downlist frames.

11.1.3.5 REFERENCES

1. GPC IPL PRD, section 3.6, page 61, THE HISAM DUMP (12/03/82), Rockwell International Corporation.

2. MG 03801 BFS PRD, section 3.7.1.7, page 28,1 HARDWARE INITIATED STANDALONE MEMORY DUMP, 0I8D, Rockwell International Corporation.

3. SPACE SHUTTLE PROGRAMS ORBITER AVIONICS SOFTWARE OPERATIONAL (OBS) DETAILED DESIGN SPECIFICATIONS (DDS), section 3.2.7.9 (FIOHISAM), 09/01/89, page 3.2.7.9-1, IBM.

4. SPACE SHUTTLE PROGRAMS ORBITER AVIONICS SOFTWARE OPERATIONAL (OBS) DETAILED DESIGN SPECIFICATIONS (DDS), section 3.3.1.5 (FCMNINIT), (09/01/89), page 3.3.1.5-1, IBM.

5. Source Code on Microfiche. Fiche name: FIOHISAM.

6. Source Code on Microfiche. Fiche name: FCMNINIT.

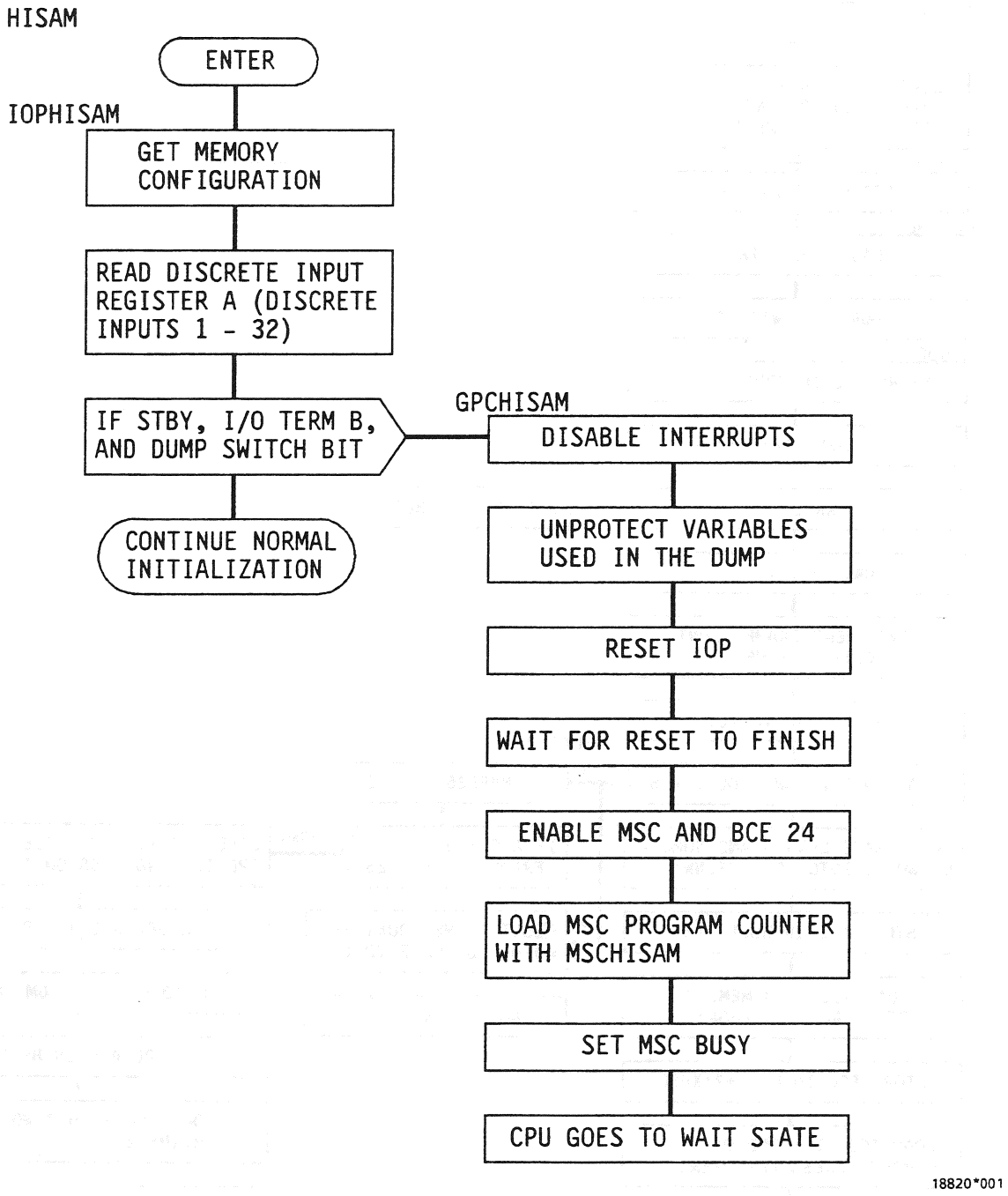


Figure 11.1.3-1.- Hardware-IPL HISAM dump program flow.

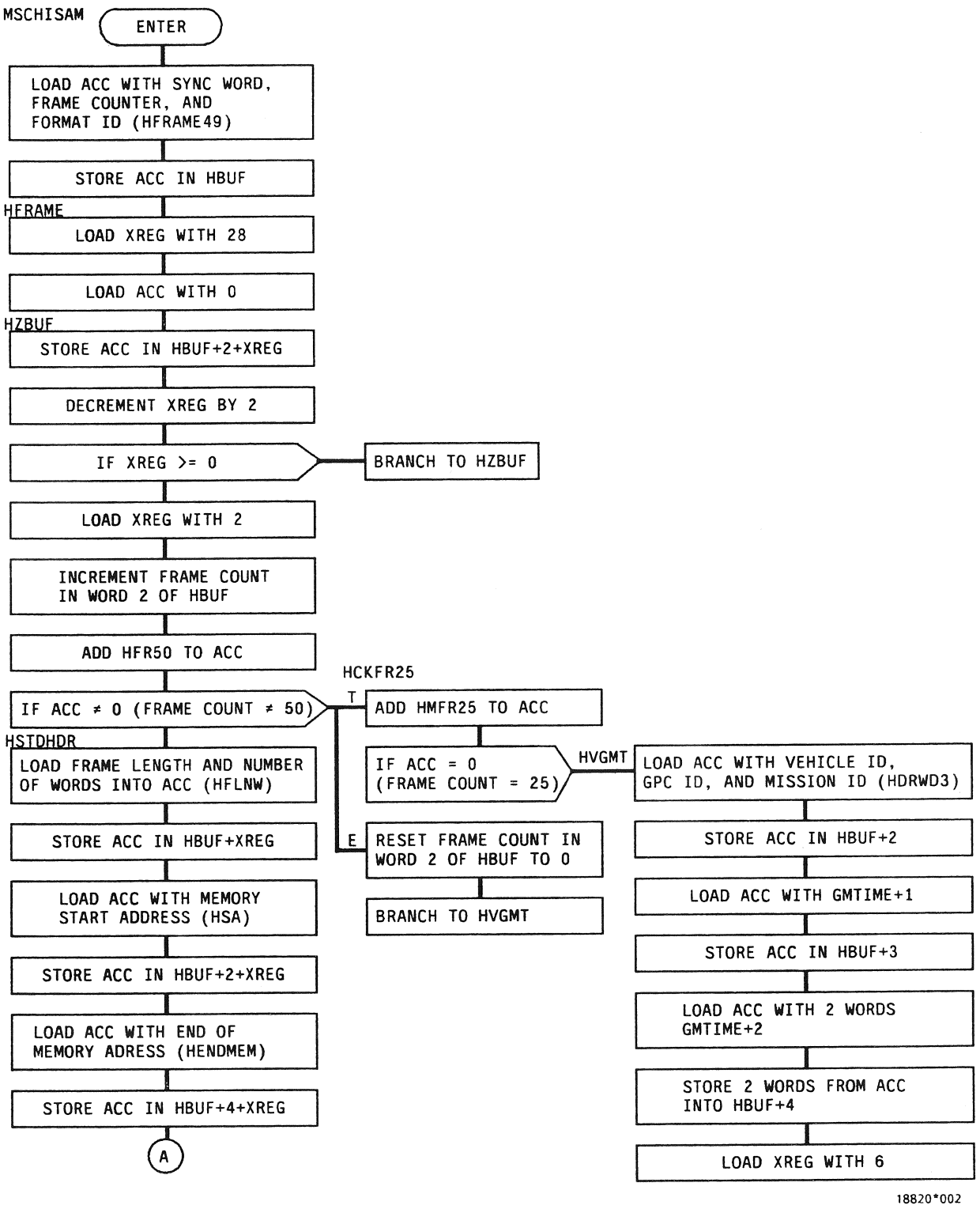


Figure 11.1.3-2.- Hardware-IPL HISAM MSC program flow.

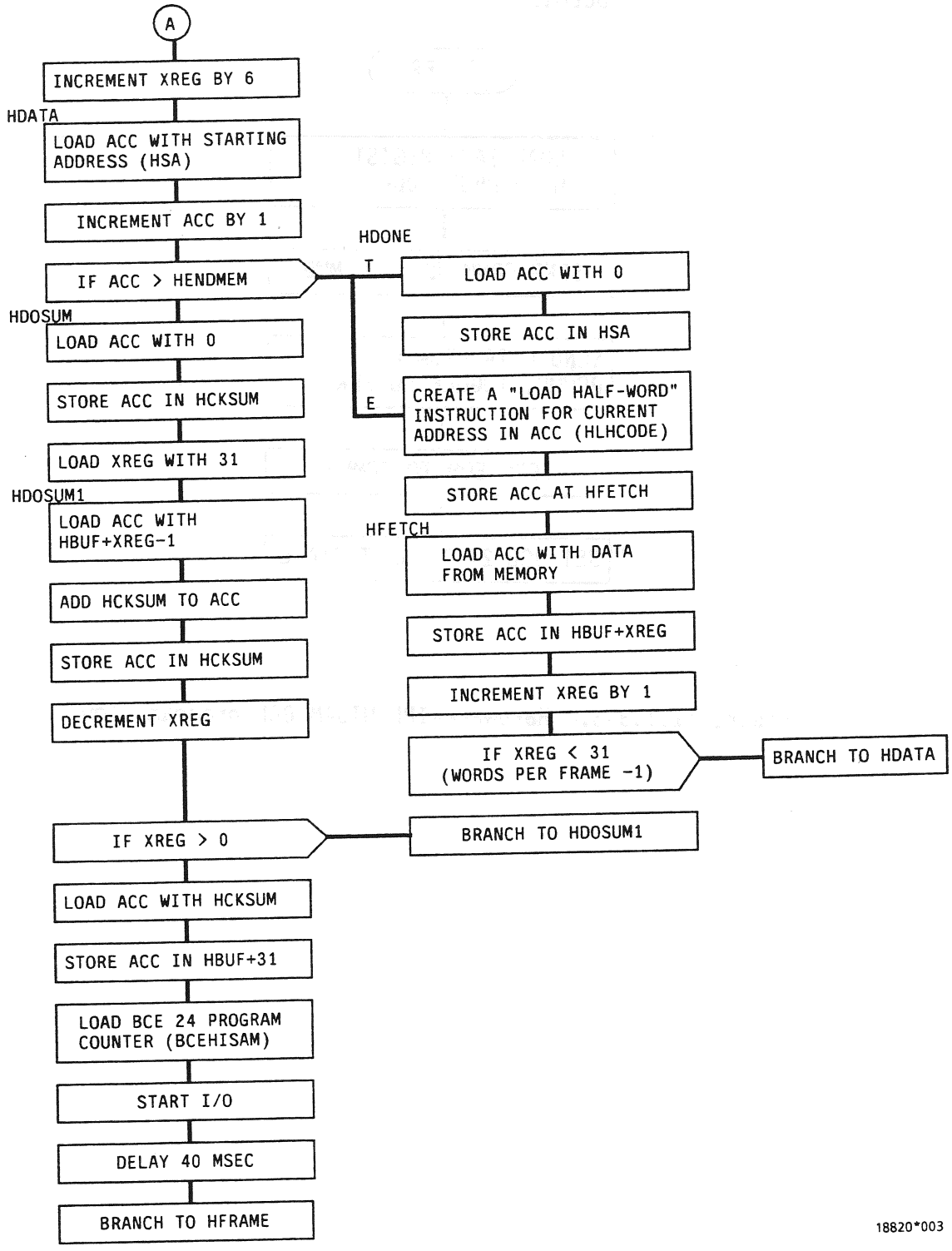


Figure 11.1.3-2.- Concluded.

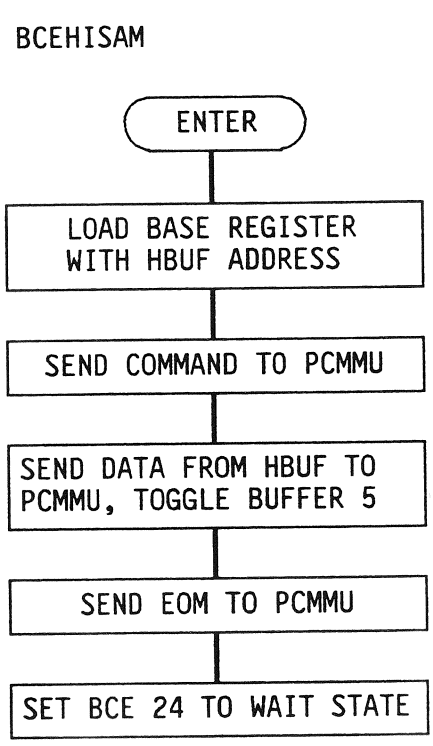


Figure 11.1.3-3.- Hardware-IPL HISAM BCE program flow.

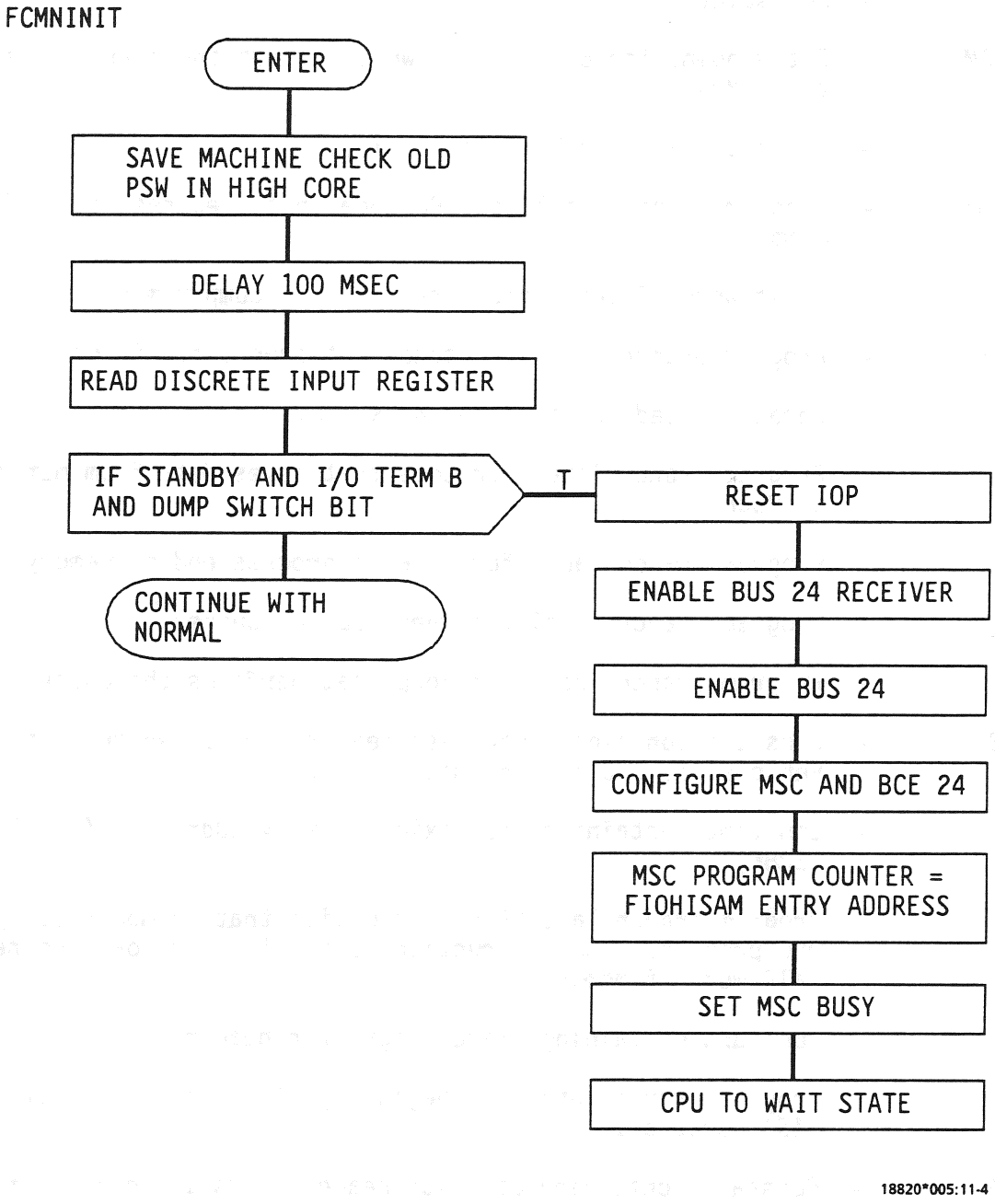


Figure 11.1.3-4.- PASS HISAM dump program flow.

TABLE 11.1.3-I.- HARDWARE-IPL HISAM DEFINITIONS

ACC	-	Accumulator.
BCEHISAM	-	Entry point for BCE program which output the buffer area to the PCMMU.
GMTIME	-	Current Greenwich Mean Time (GMT).
GPCHISAM	-	Program branch label for CPU code that prepares for a HISAM dump.
HBUF	-	First word of the buffer area used to dump memory.
HCKFR25	-	Program branch label for code that checks for frame 25.
HCKSUM	-	Location used to hold the check sum.
HDATA	-	Program branch label for code that moves data from memory to HBUF.
HDONE	-	Program branch label for code to process end of memory.
HDOSUM	-	Program branch label for check sum of buffer.
HDOSUM1	-	Program branch label for loop that performs the check sum.
HDRWD3	-	Constant containing downlist header word 3, which contains vehicle id, GPC id, and mission id.
HENDMEM	-	Constant containing the maximum memory address. (3FFFF) (128K)
HFETCH	-	Program branch label for instruction that is modified by MSC program. The instruction is modified to load the next half-word of memory.
HFLNW	-	Constant containing frame length for header.
HFRAME	-	Program branch label for beginning of infinite loop of the HISAM program.
HFRAME0	-	Constant containing downlist header words 1 and 2 for frame 0. (This will reset the frame count to 0.)
HFRAME49	-	Constant containing downlist header words 1 and 2 for frame 49.
HLHCODE	-	Op code used to build the HFETCH instruction.
HMFR25	-	Constant used to check for a frame count of 25.

TABLE 11.1.3-I.- Concluded

HMFR50	-	Constant used to check for a frame count of 50.
HSA	-	Current location in memory being dumped.
HSTDHDR	-	Program branch label for code that store the downlist header words in the buffer area.
HVGMT	-	Program branch label for code that builds long header information in frames 0 and 25.
HZBUF	-	Program branch label for loop used to zero the buffer area (HBUF).
IOPHISAM	-	Entry point for CPU program "HISAM".
MSCHISM	-	Entry point to MSC program which performs the dump.
XREG	-	X register.

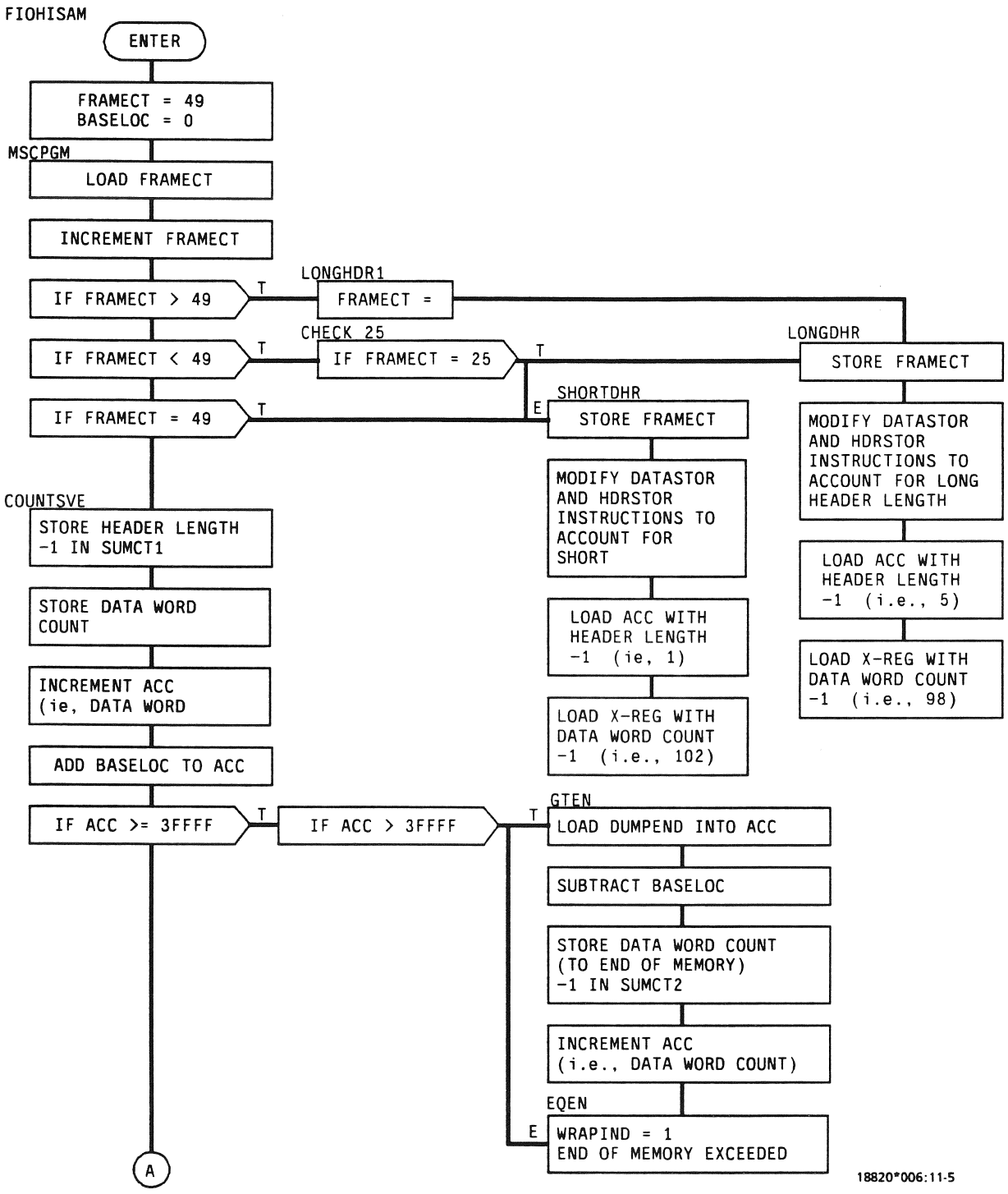


Figure 11.1.3-5.- PASS HISAM MSC program flow.

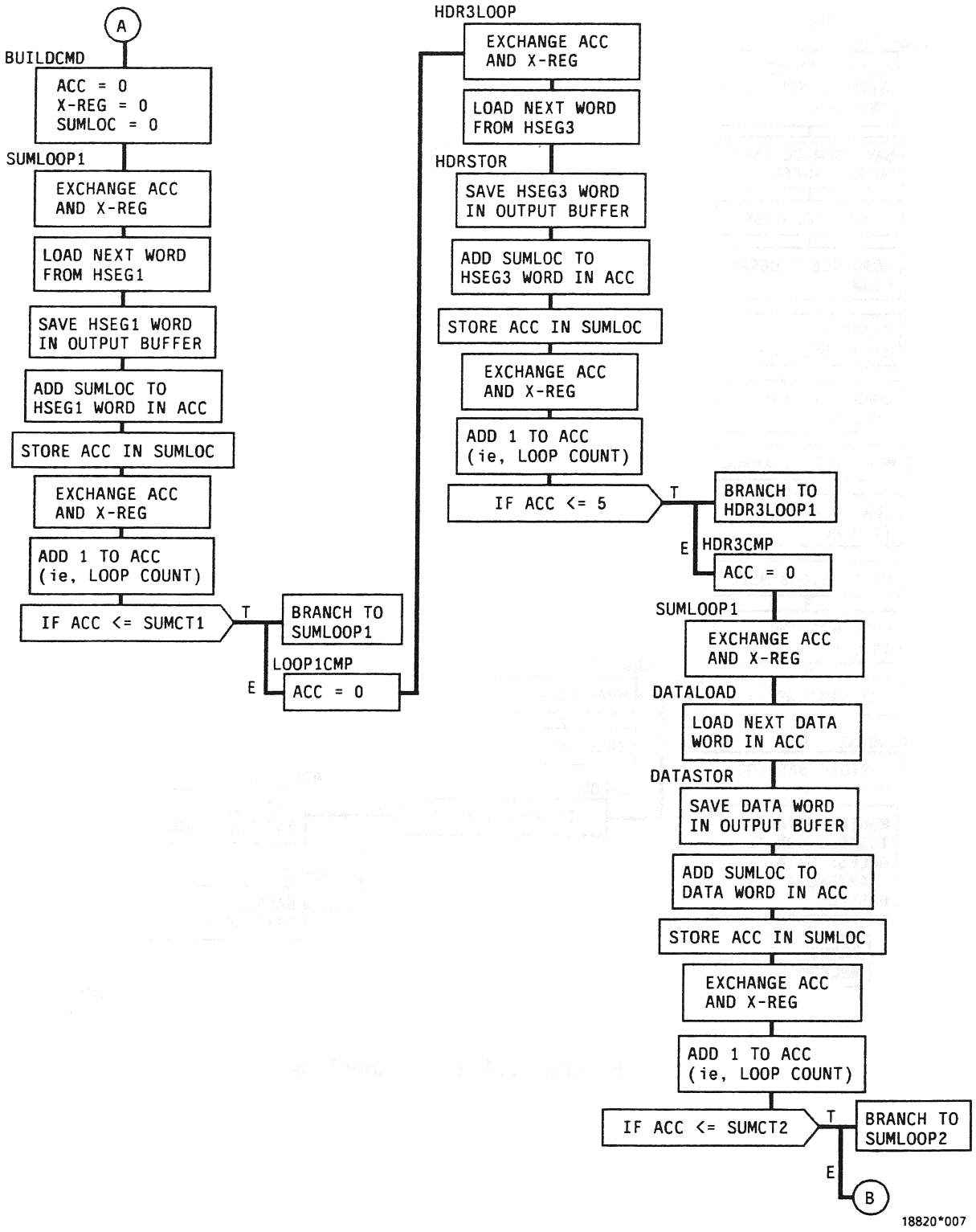


Figure 11.1.3-5.- Continued.

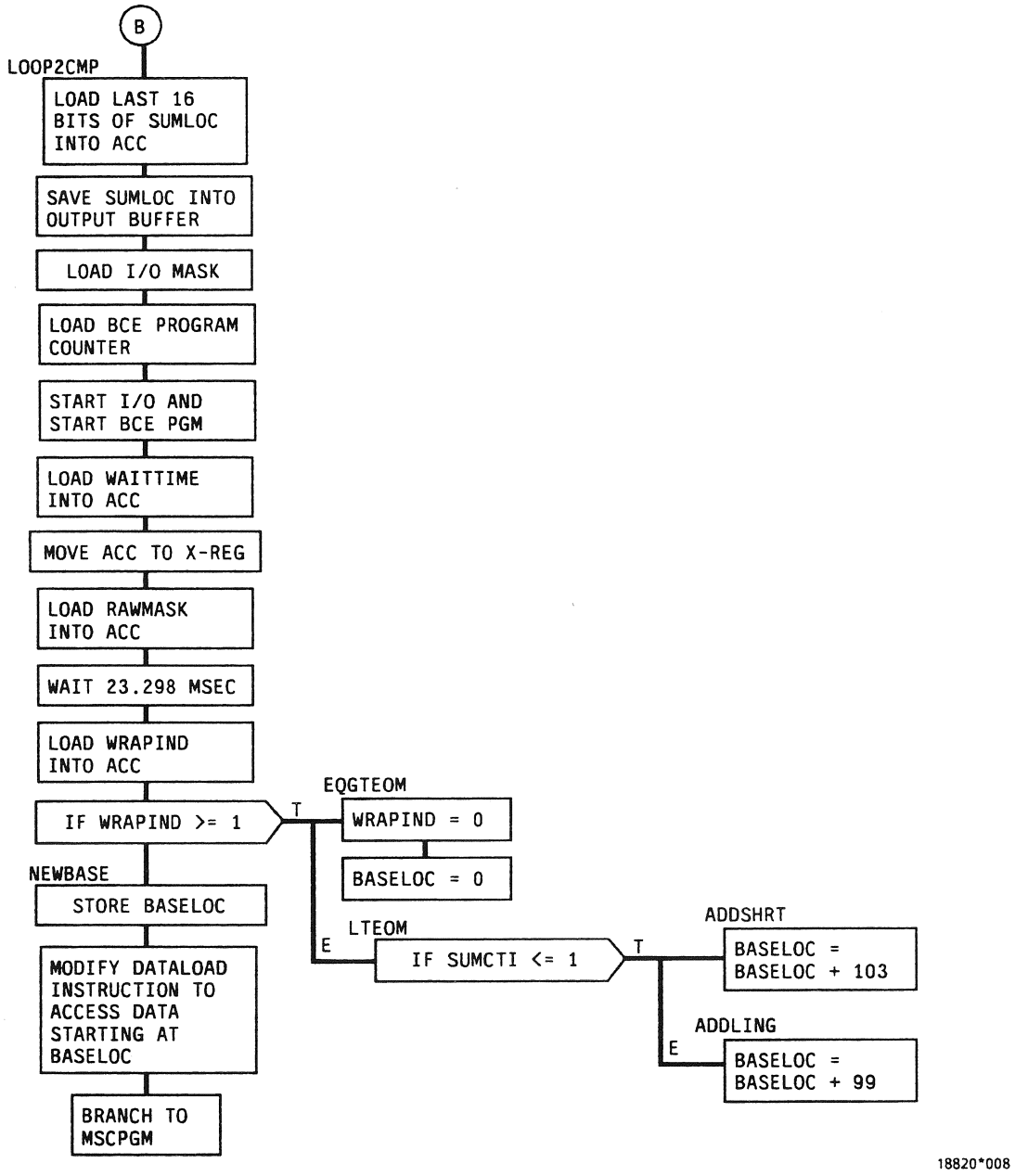


Figure 11.1.3-5.- Concluded.

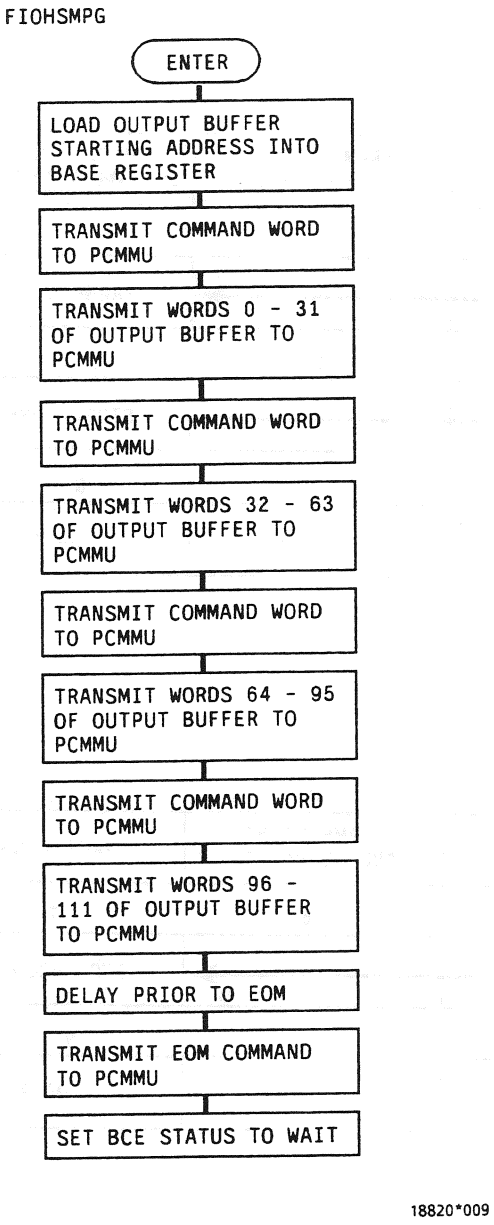
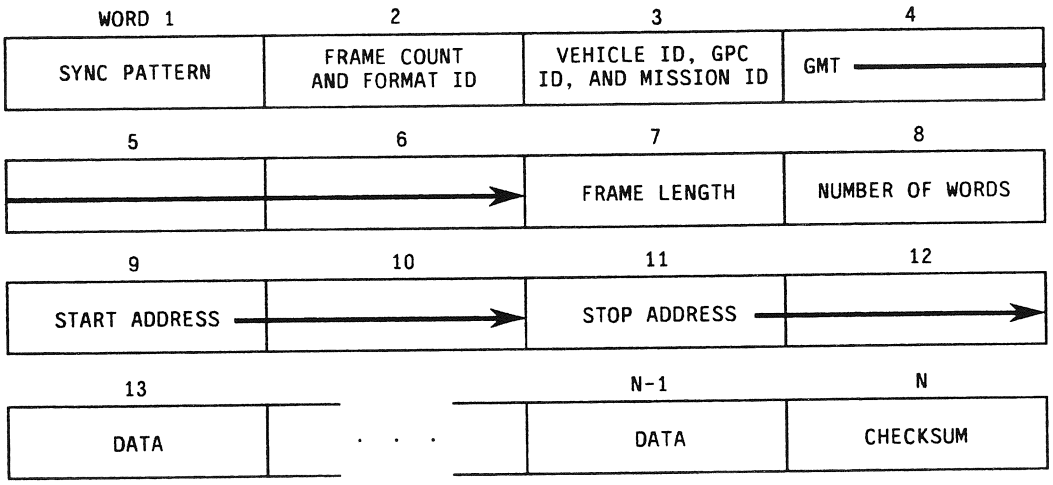
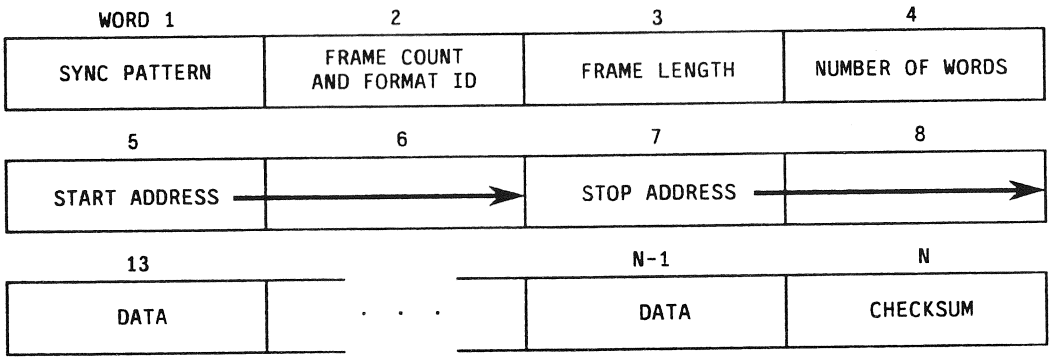


Figure 11.1.3-6.- PASS HISAM BCE program flow.

FRAME 0 OR FRAME 25:



FRAME 1-24 OR FRAME 26-49:



18820*010

Figure 11.1.3-7.- Dump formats.

TABLE 11.1.3-II.- PASS HISAM DEFINITIONS

FCMINIT	-	Normal initialization routine
FIOHISAM	-	MSC program entry point for HISAM dump
MSCPGM	-	Program branch label, for point to which the MSC program branches to begin output of next frame of dump
FRAMECT	-	Current GPC frame counter
ACC	-	Accumulator
X-REG	-	X register
BASELOC	-	Address of first memory location to be dumped in this frame
CHECK25	-	Program branch label, for code used to check for FRAMECT = 25
LONGHDR	-	Program branch label, for code used to load header words for a long header
SHORTHDR	-	Program branch label, for code used to load header words for short header
LONGHDR1	-	Program branch label, for code used to zero frame count and load long header words
SUMCT1	-	Location containing one less than the header word count
SUMCT2	-	Location containing one less than the data word count of the current frame
GTEND	-	Program branch label, for code used if BASELOC > 33FFF
EQEND	-	Program branch label, for code used if BASELOC = 33FFF
WRAPIND	-	Flag set if the end of memory has been met or exceeded
BUILDCMD	-	Program branch label, for code used to build the command word for the BCE program.
SUMLOOP1	-	Program branch label, for code used to store header words into output buffer
HSEG1	-	Memory location containing the starting address of the six header words. (The short-header will only use the first two header words.)

TABLE 11.1.3-II.- Concluded

SUMLOC	- Memory location containing the checksum so far
LOOP1CMP	- Program branch label for when sumloop1 has completed
HDR3LOOP	- Program branch label, for code used to store additional header words into the output buffer
HSEG3	- Memory location containing the starting address of the six additional header words
HDRSTOR	- Program branch label, for code that stores HSEG3 words into the output buffer
HDR3CMP	- Program branch label, for when hdr3loop has completed
SUMLOOP2	- Program branch label, for code used to store data words into the output buffer
DATALOAD	- Program branch label, for instruction used to load the next data word
DATASTOR	- Program branch label, for instruction used to store a data word in the output buffer
LOOP2CMP	- Program branch label, for when sumloop2 has completed
WAITTIME	- Number used by MSC program to wait on BCE program
RAWMASK	- Mask used by MSC program to wait on BCE program
EQGTEOM	- Program branch label, for code used to zero WRAPIND and BASELOC
LTEOM	- Program branch label, for code used to decide how much to increment the BASELOC pointer
ADDSHRT	- Program branch label, for code used to add 99 to BASELOC
ADDLNG	- Program branch label, for code used to add 103 to BASELOC
NEWBASE	- Program branch label, for code used to save BASELOC in memory
FIOHSMPG	- BCE program entry point
EOM	- End of message

11.1.4 Common and Redundant Set Synchronization

Synchronization of multiple computers onboard the Space Shuttle is the key to providing redundant, life-critical control in the very harsh environments experienced by the Space Transportation System (STS) throughout the three major phases of space flight. During the critical phases of ascent and entry, four general purpose computers (GPCs) execute identical Guidance, Navigation, and Control (GNC) software and maintain synchronization through the use of "sync codes." Sync codes are monitored several hundred times a second by all GPCs and failure to meet a sync point will cause the faulty computer to be voted out of the redundant set (RS). During the less critical orbit phase, a common set (CS) of GNC computers and a System Management (SM) computer maintain sync through the use of similar but less frequent sync codes.

The Space Shuttle is a "fly-by-wire" vehicle which cannot be controlled without the help of computers. Therefore, GNC computers in a redundant set perform the necessary GNC functions to achieve orbit, orbit, and land safely. All crew actions, transducer, and sensor inputs are fed through the computers to perform calculations and outputs necessary to control the vehicle. Failure of computer control can result in loss of crew and vehicle if the backup computer is not engaged successfully.

The solution, to provide adequate safety margins for human space flight, is the redundant computer system available in the Space Shuttle. The Space Shuttle Program terminology for the type of safety required for human space flight is "fail operational/fail safe." This means failure of one system component will result in a system which is still operational. Failure of an additional system component will result in a system which is safe but not considered fully operational.

The redundancy theory is that "identical computers receiving identical inputs and executing identical instruction sequences will compute identical outputs." The system is designed so that all computers must receive identical inputs. The computers must accept or reject the inputs which they are receiving. All computers must use identical data to perform calculations and output control commands within time and tolerance limits. Failure of a computer or computers to perform these tasks must be isolated by the remaining "healthy" computers.

A. Hardware

The GPC hardware is described in detail in System Briefs 2 and 3. In brief, each computer has three output discrete lines and twelve input discrete lines in order to maintain synchronization. This configuration allows for a total of eight sync patterns to be broadcast by each GPC. Currently, there are seven sync patterns and one spare. Moreover, the Inter-Computer Communication (ICC) buses are used to transfer data between computers in order to verify data homogeneity. Finally, there are four fail vote discretes in and four fail vote discretes out of each GPC which are used to fail faulty computers out of the set.

B. Software

1. Introduction

In general, the sync software monitors the sync patterns from the other GPCs to remain in sync and verifies data usage by the other GPCs through information received on the ICC buses (see System Brief 11.3.1). The goal of the sync software is to be transparent to application programs. However, due to the finite time required for GPC synchronization, this goal cannot be met. The sync software must always maintain two requirements if synchronization is to be successful. First, each GPC must detect and agree with the sync codes from all other members of the common and redundant sets. Second, each GPC must issue its own sync code long enough to be detected by the other set members.

a. Classes of GPC Synchronization

It is important to understand the different classes of GPC synchronization before moving on. First, the Common Set is a group of two or more computers which may have subsets of a redundant GNC set, an SM computer, an OPS 0 computer, and/or a PL computer. The "common" factor between all members of the CS is the fact that they all run the same systems software. Limited information (see System Brief 11.3.1) is passed between all members over the ICC busses and sync is maintained at 6.25 hertz. Common set GPCs are only synchronized with respect to time. Second, the Redundant Set is a group of two or more computers executing identical software. Extensive information is relayed between members on the ICC buses and sync is maintained at 300 to 400 hertz depending on what the software is executing. For example, if the GNC computers are calculating targets, the CPU usage of the calculations might be high enough to slow the RS sync down to 300 Hz. Redundant set synchronization is maintained with respect to time, data gathering, and software queue handling. Computers that are CS members but are not RS members are called simplex GPCs. Finally, a standalone computer exists by itself with no synchronization or communication with any other GPCs.

b. Sync Code Priority

The GPCs can broadcast three types of sync codes with relative priorities for each. First timer sync codes of highest priority are issued to maintain time synchronization. Likewise, I/O Completion codes are issued with a middle priority to maintain sync for I/O processes. Finally, Supervisor Calls (SVCs) of lowest priority are issued to handle synchronization of system functions. This sync code priority scheme is utilized to avoid sync failures caused by different GPCs arriving at different sync points at the same time. It should be noted that common set normal SSIP sync and redundant set timer sync are timer sync codes of the same priority. Hence, to avoid contention, the

SSIP sync is initiated by a timer interrupt, not a dedicated SSIP interrupt.

TABLE 11.3.2-1.- SYNCHRONIZATION CODES[2]

Decimal	Binary	Purpose
0	000	Halt/Off In-Sync For All Routines
1	001	I/O Complete, no error (IOC)
2	010	I/O Complete, with error (IPR)
3	011	SPARE
4	100	SVC Interrupt
5	101	Timer Interrupt
6	110	SSIP Interrupt
7	111	Null/Run

c. Sync History Log

The GPC software maintains a sync history log in order to provide information for analysis of GPC sync failures. Sync history entries are logged by SVC_Synchronization_Processor for SVC sync failures, I/O_Synchronization_Processor for I/O sync failures, and TQE_Expiration_Processor for timer sync failures. Besides sync trace logs entries for fails-to-sync, sync trace log entries are made by SVC_Synchronization_Processor and I/O_Synchronization_Processor whenever they are interrupted for higher priority processing.

Unless sync trace entries are made by early suspension of low priority sync processing or by a GPC being manually moded to HALT, STBY, or OFF, the sync history log is "frozen" when a GPC fails-to-sync and can only be reinitialized by an Item 48 on SPEC 0. The sync trace log is stopped by zero'ing out a modification word after incrementing the log entry number by one after a sync fail. Since the modification word is zero, all entries into the sync trace log after a fail-to-sync are made to the same log location. After an Item 48 on SPEC 0, the modification word is incremented to allow further sequential entries into the sync trace log to be made.

2. Synchronization Failure Modes

There are many manners in which GPCs can fail-to-sync. Some of these failures are described below to exemplify the complexity of the synchronization process. It is important to note that, although an RS or CS fail-to-sync is generally thought of as a single GPC failing, multiple GPCs can fail as well. Multiple GPC failures in the RS are generally classified by the number of GPCs in discrete sets or standalone after the failure. For example, a "one-on-one-

on-one-on-one" is a total redundant set split or failure. Likewise, a "two-on-two" is a failure of the four GPC RS into two, two GPC sets. Finally, a "two-on-one-on-one" is a situation where two GPCs fail out of the RS and do not maintain sync with one another.

There are two types of fail-to-sync which must be understood from a fundamental standpoint. GPCs fail-to-sync and force-fail-to-sync. Moreover, it is important to note that a GPC does not care who votes against it, only who it votes against. In other words, input fail votes are only handled in the IOP hardware; whereas, output fail votes must be calculated based on perceived error conditions.

A fail-to-sync results when a GPC does not meet a sync point. All GPCs which detect that a GPC has not established sync within the allotted time window will call the sync fail routine to mask the buses of the GPC which did not meet the sync point, and the voting GPCs will freeze their sync trace logs. Fail votes will be issued against the offending GPC, and the failing GPC's I-fail will be lit if two or more GPCs vote against the offender. If the RS is equal to the CS, then an RS FTS will result in a CS FTS. However, if the RS is a subset of the CS, a GPC can fail RS sync but maintain CS sync.

A force FTS results when a GPC recognizes that only it has experienced an error. The actual logic involves an "everyone else is bad, so I must be bad" approach. RS or CS force FTS in this scenario requires an RS or CS set with at least three members. The faulty GPC will set its own I-fail, freeze its sync discretes at a non-000 value, and freeze its sync trace log. At this point, the failing GPC still has control of all the buses which it was assigned. Due to the fact that the failing GPC's sync discretes are frozen at a non-000 value, it will miss the next sync point. This will result in sync fail processing by the remaining GPCs in the manner described in the previous paragraph. A GPC will force FTS if it encounters a nonuniversal I/O error, it is the only GPC reporting an error, multiple GPCs have errors on their ICC bus (bus commander fails), or multiple GPCs report errors but do not agree on bus (reporting GPC fails). Besides the I/O error Force FTS scenario, other errors such as parity errors between and within the IOP and CPU will cause a GPC to force FTS.

For an FTS or force FTS in a redundant set of three or four GPCs, if the failing GPC is the downlister, then the sync fail processing software will cause the failing GPC to issue an illegal interface unit address (IUA) to the PCMMU. This will cause the lowest ID GPC in the RS to assume the downlist task.

For an RS of two GPCs, the force fail logic will not work. Instead, each GPC will mask each other's buses, I-fails will not be issued, and sync lines will not be frozen.

a. "Sync Hole"

Synchronization is accomplished by each GPC examining the sync codes of the other GPCs. This function is implemented by a sync routine which has a window of 3.85 milliseconds in which the GPC will wait for the other GPCs to produce the proper sync code. Failure of a GPC to establish sync within this window will result in a "fail-to sync" of the faulty GPC. However, if the set of GPCs experiences a nonuniform perception of sync codes, the result can be a breakup of the redundant or common set. This type of error is referred to as a "sync hole."

b. "Data Homogeneity"/Protected Transactions

Another type of failure which can cause GPCs to fail-to-sync is nonhomogeneous data. The Shuttle software architecture is based on asynchronous, priority driven processes. As a result, variables which are shared between processes, interprocess variables (IPVs), must be protected from the interruption of processes which could change IPVs differently in different GPCs. If this happens, divergent processing may result in an eventual fail-to-sync.

To protect IPVs from interruption problems, several variable protection schemes are utilized by the PASS software.

First, GPCs can be synchronized and then interrupts can be disabled while IPVs are accessed. After IPVs have been accessed, the interrupts are enabled. The advantage of this approach is automatic verification during the software build process. The disadvantages of this implementation are an increase in the sync overhead and irrelevant entries in the sync trace log.

The second method of IPV protection involves the use of mutually exclusive processes. This method of IPV protection involves forcing processes which use IPVs to be executed at different times. This process can only be utilized if the rates of the processes involved do not overlap. The other disadvantage of this method is that it requires a desktop analysis of the IPVs involved to ensure correct access of the IPVs by the involved processes.

Finally, a method of data encapsulation can be employed to protect IPVs. As the name of the method implies, this type of protection eliminates IPV problems by effectively eliminating IPVs. The implementations of data encapsulation include the use of dedicated variable copies, shadow variables, and exclusive procedures. Shadow variable implementation involves a disabling

of interrupts while the IPVs to be utilized are moved to a local variable copy for processing. Then, interrupts are enabled and processing continues. When the modifications are completed to the local variable copies, the results are stored in the permanent variables. Exclusive procedures are used to centralize the access to a set of IPVs. This method suspends low priority processing of an exclusive procedure if a high priority interrupt is issued. However, if the high priority process then tries to use the same exclusive procedure, the high priority process is suspended while the low priority process completes the exclusive procedure. Control is then passed back to the high priority process. The main concerns of this implementation are the amount of disable time needed to transfer variables and restrictions placed on the exclusive procedures. Shadow variable and exclusive procedure implementation can both be verified automatically during the software build process.

c. Direct Memory Access Interference

Direct memory access (DMA) interference between the GPC's CPU and IOP can also cause a GPC to fail-to-sync. This condition can result if a GPC in the redundant set is performing an inordinate amount of HFE I/O processing relative to the other members of the set. The overloaded GPC is slowed down by the amount of DMA activity which it must perform and as a result misses its sync point. The sync code is written to handle a 4/0 string imbalance in a two GPC set and a one string imbalance in a three or four GPC set.

A commonly used example of this failure mode is referred to as the 2/0 string imbalance. This refers to a condition in which one GPC in the RS has two flight-critical strings assigned to it while another member of the RS has no flight critical strings. This imbalance of processing activity may cause the GPC with two strings to fail-to-sync.

d. Skew

Although the GPCs' internal clocks are synchronized every 960 ms with the MTU, factors such as runtime differences, I/O completion time, and internal oscillator differences can cause enough time skew to cause a fail-to-sync. However, the sync code is written to take into account the majority of these problems.

e. ICC Checksum Miscompares

Details of the checksum software processing can be found in System Brief 11.3.1. The following is only a description of the manner in which checksum miscompares may cause an FTS.

If a checksum miscompare occurs, a retry of the compare will be processed. Then, if there is another consecutive checksum miscompare for the variable ICC buffer data, the GPC which is different than the rest will force itself to fail-to-sync. A GPC error code of 05 07 will be logged. The variable portion of the ICC buffer is checksummed due to the fact that the variable data is critical at the time it is sent.

A sumword miscompare of the fixed ICC buffer data will not result in a fail-to-sync. The fixed ICC buffer data sumword miscompare check is performed once every four major cycles. After three consecutive sumword miscompares (11.52 seconds), a SUMWORD ICC fault message will be issued. A GPC fail-to sync may not occur as it does with the variable ICC buffer. If the miscompares continue, the fault message will not be reissued but will only be issued once. Even though this failure does not cause a fail-to-sync, the reason it is mentioned is because there is a CR pending for OI-20 to cause this particular failure to result in a fail to sync.

f. Flight Critical Bus Transmitter/Receiver Failures

GPC transmitter/receiver (T/R) failures can cause fail-to-syncs due to the commander/listener logic utilized by the PASS RS GPCs. If the receiver of a GPC fails, a nonuniversal I/O error will result and that GPC will force FTS. If the transmitter of a commanding GPC fails, the commanding GPC will log an initial timeout while the listening GPCs will log an MSC timeout. If the receiver of the same BCE is also failed, that GPC will experience a nonuniversal I/O error if restring is attempted resulting in a force FTS.

g. PASS Program Note References[3]

DR025159 OVERLAY FAILURE FOLLOWING F-T-S DURING OVERLAY
DR025187 F-T-S DUE TO PCMMU DATA
DR028350 POSSIBLE FAIL-TO-SYNC DUE TO LAB ERRORS AND ICC OVERLOAD
DR029284 SPEC 002, GMT UPDATES
DR030138 RAPID CS FORMATION
DR032119 OUTPUT SWITCH IN TERMINATE AT RS FORMATION
DR037536 RESTRICTION ON SYSTEM SOFTWARE SPEC USAGE
DR037543 GPC FAILED OUT OF COMMON SET/REDUNDANT SET
DR037590 MULTIPLE COMMANDERS OF A DEU CAN RESULT IN AN FTS
DR038458 DUAL COMMANDERS AFTER LOSS OF RECONFIGURATION ICC MESSAGES
DR042433 POTENTIAL F-T-S DUE TO ICC CONTENTION AT NEW GPC START UP
DR052100 BUS DISTRIBUTION RESTRICTIONS
DR055017 EXECUTING SACS OPERATOR ACROSS OPS TRANSITION
DR055300 INCONSISTENT ANNUNCIATION OF FTS DURING OPS TRANSITION
DR055318 POSSIBLE COMMON SET FTS DUE TO I/O COMPLETION
DR101410 MDM HW FAILURE MAY CAUSE CONTINUOUS I/O ERRORS AND RS FTS
DR101691 SPEC2 ITEM ENTRIES MAY CAUSE FTS AT OPS TRANSITION
DR103127 POSSIBLE FTS WHEN PERFORMING SSME LOAD AND TELEMETRY FORMAT LOAD
DR103130 POSSIBLE FTS WHEN TERMINATING IN SPECS 101 AND 102 IN G9

3. Software Modules [1]

Details of all of the software modules discussed here can be found in part 1 of the Systems Software Detailed Description Document section 3.3.5, GPC Redundancy Management. Included are module descriptions, input descriptions, output descriptions, and flowcharts.

There are five software modules which contribute to the maintenance of common and redundant set sync.

- Initial_SSIP_Synchronization_Processor (FCMASYNC) - special
- Normal_SSIP_Synchronization (FCMCSYNC) - CS
- I/O_Synchronization_Processor (FCMISYNC) - CS and RS
- SVC_Synchronization_Processor (FCMSSYNC) - RS
- Timer_Synchronization_Processor (FCMTSYNC) - RS

Over and above these normal synchronization modules, there are eight other modules which handle the processing of fail-to-syncs. A brief description of these modules is given here.

- Sync_Fail_Processor (FCMSFAIL) - DDS 3.3.5.6
  Performs the processing necessary to handle force FTS and FTS.

- Sync_Fail_CAM_MSC_Processor (FCMSFCAM) - DDS 3.3.5.7
  Handles the issuing of fail discretes for FTS situations

- Fault_Detection_Identification (FCMDFI) - DDS 3.3.5.8
  Performs sumword comparison processing.

- Fail_Discretes_MSC_Processor (FCMSVOTE) - DDS 3.3.5.9
  Handles the issuing of fail discretes for modes to STBY and HALT.

- GPC_Discretes_Redundancy_Manager (FCMDSCRM) - DDS 3.3.5.10
  Performs the GPC discrete RM processing. All GPC discretes are compared and a CS RM result is provided.

- Sync_Mask_Build_Routine (FCMSMASK) - DDS 3.3.5.11
  Service routine used by all sync routines.

- Sync_Fail_Interface_Routine (FCMSFINT) - DDS 3.3.5.12
  Builds a parameter list necessary for the Sync_Fail_Processor.

- Sync_Trace_Routine (FCMTRACE) - DDS 3.3.5.13
  Logs sync trace entries.

a. Initial_SSIP_Synchronization_Processor (FCMASYNC) - special
   (Refer to DDS section 3.3.5.1 for more details.)

   The Initial_SSIP_Synchronization_Processor (FCMASYNC) is a special process designed specifically to assist the synchronization of a GPC not in the common set with the currently existing common set members. If there are no current common set members, no sync is necessary. Upon entering FCMASYNC, FCMASYNC will not try to establish sync initially. This is done in order to ensure that the GPC which is entering the common set will be able to recognize all of the members of the common set. If the GPC entering the common set tried to establish sync on the first time through and the existing common set was executing SSIP sync, the trailing edges of the changing sync discretes might not be fully interpreted by the GPC trying to establish sync. The result would be a GPC which would only attempt to establish sync with part of the true common set.

b. Normal_SSIP_Synchronization (FCMCSYNC) - CS
   (Refer to DDS section 3.3.5.2 for more details.)

   Normal_SSIP_Synchronization (FCMCSYNC) is the process designed to maintain common set sync and monitor for GPCs which may be trying to enter the common set.

Common set sync is maintained by issuing the SSIP sync code (110 binary) at every fourth SSIP sync point. Each GPC attempting to establish sync with the other members of the set will wait to recognize the SSIP_Sync_Code of the other GPCs for an amount of time determined by the PC1 timer which is set to 3.85 ms. When the time expires for every GPC to establish sync, the GPCs which did not receive the sync codes which they expected will call the Sync_Fail_Processor to eliminate from the set the GPC or GPCs which did not meet proper sync. It is important to note that the loop timer (PC1) is reset every time a GPC syncs with others to ensure that all GPCs leave the sync loop at the same time.The common set sync and the redundant set sync algorithms are very similar. The only difference is the sync code which is searched for by the GPCs trying to establish sync. Common set sync looks for a sync code pattern of 110, and redundant set sync looks for a sync code pattern of 101. The details of this algorithm are presented in the description of the Timer_Synchronization_Processor. Refer to figure 11.1.4-1, Generic Common/Redundant Set Sync Algorithm.

GPCs eligible to enter the common set are monitored for a sync pattern of binary 000. As long as current members of the common set consider a GPC, which is not in the common set, eligible to join the common set and a sync code of binary 000 is received from that eligible GPC, then the active GPCs will continue to monitor the eligible GPC for the conditions necessary to enter the common set. The active GPCs must detect a Null/Run sync code on (logical 1) during one processing cycle and a SSIP sync code on (logical 1) the next processing cycle when an eligible GPC is taken to RUN in order for that GPC to sync up with the common set.

c. I/O_Synchronization_Processor (FCMISYNC) - CS and RS
(Refer to DDS section 3.3.5.3 for more details.)

I/O synchronization for the common and redundant sets is accomplished through the use of the I/O_Sync_Code and the IPR_Sync_Code. The logic of the I/O sync is similar to the Generic Common/Redundant Set Sync Routine (fig. 11.1.4-1) with two major exceptions. First, since I/O sync is of lower priority than timer sync, timer interrupts are enabled and disabled every time through the I/O sync check loop to allow timer sync to occur. Second, the sync trace log is utilized to log entries when the PC1 timer is expired. As will be discussed in detail in the timer sync description, the sync discretes are read twice and logical OR'ed together to ensure the correct discretes are considered.

Moreover, the PC1 timeout counter is reset every time a GPC establishes I/O sync.

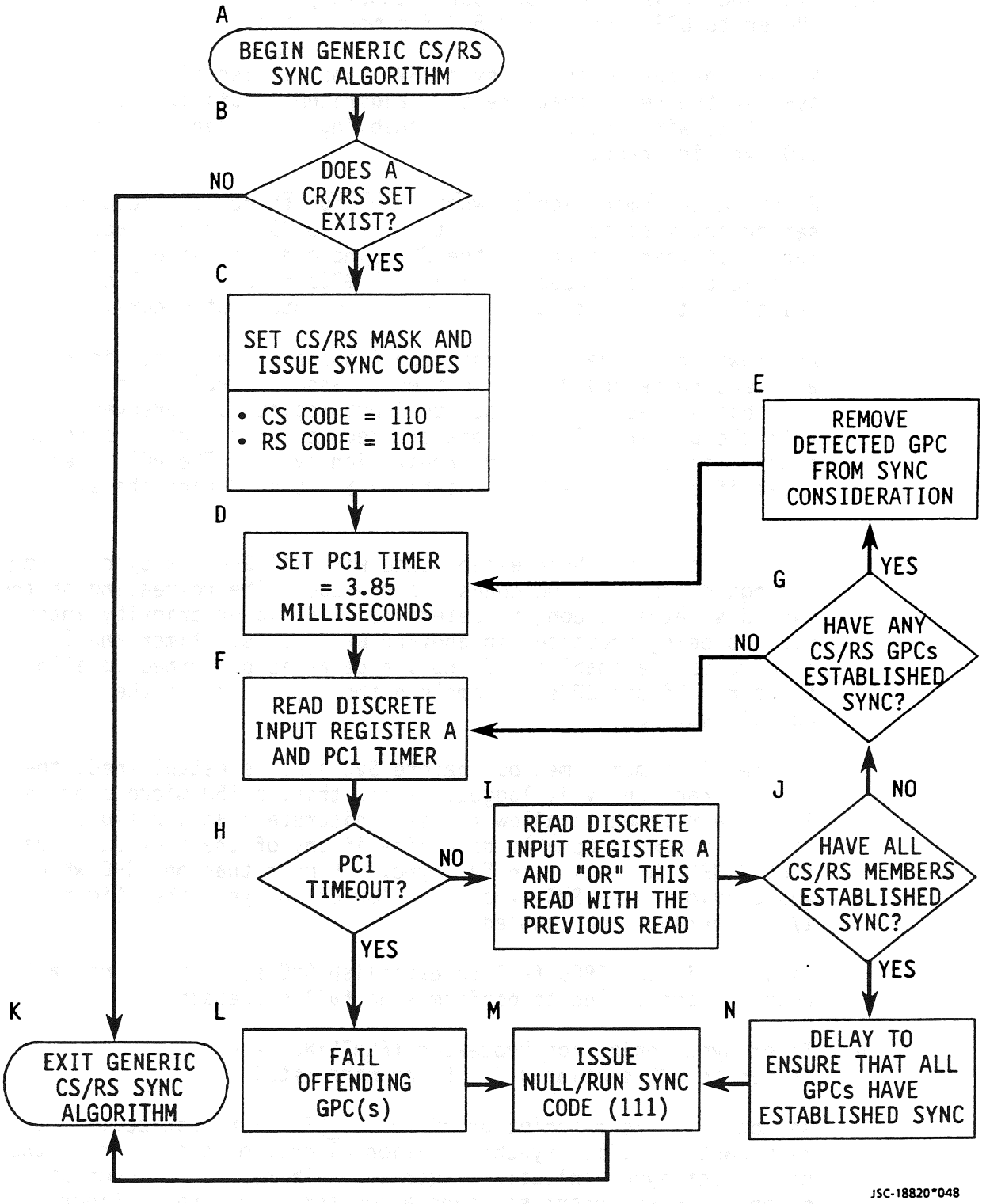


Figure 11.1.4-1.- Generic common/redundant set sync algorithm.

d. SVC_Synchronization_Processor (FCMSSYNC) - RS
(Refer to DDS section 3.3.5.4 for more details.)

As was the case with I/O sync, SVC sync is also similar to timer sync in the sense that the same algorithm is utilized (see fig. 11.1.4-1) with the addition of enabling and disabling timer and I/O sync interrupts.

First, a determination of whether or not there is a redundant set or not must be made. If there is no RS, a sync trace log is made. If there is an RS, the SVC sync code is issued. The SVC sync loop is continued until all RS GPCs establish SVC sync, the PC1 timer times out, or a timer or I/O interrupt occurs.

As shown in the generic sync loop flowchart, the sync discretes are read twice and OR'ed together to assure stability of the sync bit values in the discrete input register. Moreover, GPCs with the proper SVC sync code are removed from further sync consideration during this synchronization cycle. The PC1 timer is reset if one or more GPCs establish SVC sync during the sync loop.

When SVC sync has been established with all GPCs, a sync trace is logged and the sync codes are re-read. The re-reading of the sync discretes is done to determine if a higher priority interrupt is being processed in another GPC. If so, timer and I/O interrupts are enabled. If not, a delay is performed to allow the other RS set GPCs to complete their SVC sync if they have not already done so.

If the PC1 timer times out before SVC sync is established, then a sync trace entry is logged. After this, a 150-microsecond delay is performed to allow for sync discrete stabilization. Then, a check is done to determine if any of the previously detected GPCs are still in SVC sync. If more than one GPC which was previously in SVC sync is no longer in sync, then timer and I/O interrupts are enabled.

Finally, if any GPCs fail to establish SVC sync, the sync fail routines are called to perform sync fail processing.

e. Timer_Synchronization_Processor (FCMTSYNC) - RS
(Refer to DDS section 3.3.5.5 for more details.)

As noted in the description of Normal_SSIP_Synchronization, the redundant set timer synchronization algorithm is similar to the common set synchronization algorithm. Therefore, a description of only the redundant set sync algorithm according to figure 11.4.1-1 will be given here.

First, a determination of whether or not a set exists with which to sync is performed (block B). If so, the redundant set sync

code of 101 is issued (block C). Next the first of two loops is begun by setting the PC1 timer to 3.85 ms (block D). The discrete input register A (DIR-A) and the PC1 timer is then read (block F).

If the 16-bit PC1 timer register is equal to 1, 3.85 ms has expired without a sync code being received from some or all of the GPCs in the set (block H). The GPC(s) which do not meet the sync point are voted against (block L).

If the PC1 timer has not expired, the DIR-A is read again (block I). A logical OR is performed with this read of DIR-A and the previous read of DIR-A in block F. At this point, if the proper sync code from all of the members of the set have been detected (block J), then a short delay is processed to ensure that all of the GPCs in the set have completed sync processing (block N).

If all of the members of the set have not met sync (block J), then a check is performed to determine if any GPCs have met sync during the current cycle of the inner loop F-H-I-J-G (block G). If not, the inner loop is begun again to look for sync codes as the PC1 timer is still counting down.

If at least one GPC establishes sync during the inner loop, that GPC is removed from further consideration as a GPC which must meet sync (block E). Then, the outer loop of D-F-H-I-J-G-E is begun again. Notice that the PC1 timer is reset every time a GPC establishes sync. This ensures that all GPCs which have met sync will have minimum sync skew. This outer/inner loop cycle is continued until all in the redundant set have established sync with each other.

C. References

1. Detailed Description Document, Part 1, Flight Computer Operating System (FCOS), April 2, 1982, section 3.3.5, GPC Redundancy Management.

2. SSW Maintenance Specification Document, FSW REL OI-8B, September 3, 1987, SSW On-Board Shuttle Systems, MC 8212, IBM Corporation, 3700 Bay Area Boulevard, Houston, Texas 77058.

3. Addendum 5, OI-81C Flight Software Program Notes and Waivers, August 24, 1989, published March 16, 1990.

11.1.5 G3 Archive

11.1.5.1 Background

The G3 archive is a redundant source of entry software loaded into the upper memories of the PASS GPC's. Two item entries on the DPS Utility display, SPEC 1, control the G3 archive function (fig. 11.1.5-1). Item 48 enables the loading of the G3 overlay into the upper 128K of GPC memory on a G9-to-G1 OPS transition. Item 49 enables the retrieval of the software from upper memory and places it in lower memory for usage upon a subsequent G3 transition. These item entries are mutually exclusive of one another. In other words, both functions cannot be enabled at the same time. The following systems briefs will go into depth on the load, retrieval, initialization processes, and the DR's that have been written against this software capability.

11.1.5.2 G3 Archive Load

The G3 archive load function, an SSW process, provides the ability to load the G3 overlay into upper GPC memory (128K-256K). Item 48 on SPEC 1 can enable/disable the G3 load function only in G9. The execution of this entry in any other operational sequence will result in an ILLEGAL ENTRY message. Also, enabling this item entry will not initiate the load; a G9-to-G1 transition is required for the load to take place. Therefore, no capability exists to perform a G3 archive load on orbit.

The following is a sequence of events that occurs during a G3 archive load: item 48 (load) enabled, automatically enabled upon primary GPC initialization, followed by a G9-to-G1 transition:

OPS 101 PRO

A. The OPS 3 overlay is loaded from the MMU into the lower memories of the GPC's targeted for MC 1.

B. The overlay goes through a store protection process so that it will not be overwritten unintentionally in upper memory.

C. The overlay is copied to the upper memory and "packed" to minimize retrieval time. (The term "packed" means the G3 software is stored contiguously in upper memory. In other words, there are no gaps between the memory locations where the archive exists.)

D. The OPS 1 overlay is loaded into all GPC's targeted for MC 1, overwriting the G3 overlay in the lower memories.

NOTE: The OPS 1 transition time will approximately double in length when a G3 archive load is done.

If all targeted GPC's obtain the G3 overlay, the G3 load function is successful. Item 48 will automatically disable, and item 49 will automatically enable. If the G3 load function is unsuccessful (not all GPC's received the archive), then item 48 will remain enabled, and item 49 will remain disabled. For additional information on the unsuccessful/off-nominal loading of the G3 archive refer to section, 11.1.5.2.1.

On the ground, the G3 status from each GPC is seen on MSK 1302, DPS 2 GNC/BFS, under G3 ARCHIVE (fig. 11.1.5-2). Each represents the common set information of each GPC status, 1 to 5 from left to right. A "0" indicates no archive is present in the GPC and a value of 1, 2, or 3 indicates which G3 software copy was loaded from the MMU.

11.1.5.2.1 Off-nominal.-

A. Unsuccessful G3 archive load

As with other OPS transitions, if any GPC is unable to obtain the software (G3 archive) from the MMU (1/2), then SSW will automatically switch to the other MMU (2/1) and retry the transaction. If not all GPC's acquire the archive on the second try of the alternate MMU, then the G3 archive load is considered unsuccessful (item 48 remains enabled, and item 49 remains disabled). However, unlike other transitions, these mass memory I/O errors will not cause the erring GPC(s) to fail to sync. In essence, nonuniversal I/O error processing is "suspended" during this period. Yet, the fault summary messages will still be annunciated. Upon the OPS 1 transitions, the nominal PASS error processing will be reinstated.

B. No I/O errors, but a GPC without archive

If no I/O errors are logged during a G3 archive load, but a GPC did not get the archive, the GPC is considered suspect. The reasoning behind this judgment is that the GPC probably obtained the G3 overlay successfully in lower memory. However, upon the transfer of overlay to upper memory, a problem occurred either in software/hardware internal to the GPC. This problem has the same signature for an unsuccessful G3 archive without the I/O errors.

C. G3 archive load successful then a recycle back to G9

If the G3 archive is successfully loaded (item 48 disabled and item 49 enabled) and the GPC's were moded back to G9, the G3 overlay will not be reloaded on the next G1 transition. However, if item 48 is selected again in G9, then G3 will be reloaded. Also, if the first G3 archive load was unsuccessful (no toggle between item 48 and item 49), then moding to G9 and back to G1 will reload the G3 overlay.

11.1.5.3 G3 Archive Retrieve

The G3 archive retrieve, a SSW process, provides the ability to retrieve the G3 overlay from the GPC upper memory and load it into the lower memory. Item 49 on SPEC 1 enables or disables the retrieve function (fig. 11.1.5-1). This item entry is only valid in G1, G2, G8, and OPS 0 and if at least one GPC contains the G3 archive. Otherwise, an ILLEGAL ENTRY message will be annunciated. Also, enabling this item will not automatically retrieve the overlay. In order to begin the transfer from upper memory to lower memory, item 49 must be enabled, and a G3 OPS transition must be done.

NOTE: The requirement that an illegal message will appear when no GPC's contain the G3 archive when an item 49 is executed does not imply that item 49 will automatically disable whenever no GPC's contain the G3 archive overlay. An example is if only one GPC holds the archive and it is re-IPLed. This IPL would erase the archive from the GPC's memory, but item 49 will still remain enabled until the operator disables it (because of CS perception of item 49 status).

However, even with the item 49 enabled and a G3 transition, PASS systems software will first check if a G3 overlay resides in any GPC lower memory. If the overlay is not present, PASS will try to retrieve the G3 archive. In turn, if no archive exists in any of the GPC's, the overlay will be obtained from the mass memory. Figure 11.1.5-3 represents this sequence of events.

The portion enclosed by dotted lines on figure 11.1.5-3 represents the G3 archive retrieve logic. The retrieve logic functions as follows:

A. If all the target GPC's contain identical G3 overlays in upper memory, then a "fast" transition is performed. In this case, all the GPC's simultaneously copy the G3 archived overlay from upper memory to lower memory.

B. If at least one but not all GPC's in the target set contain the G3 archive, then a "move and transfer" OPS transition is performed. Here, all the GPC's with the archive simultaneously copy the overlay from upper memory to lower memory. Then the lowest ID GPC containing the G3 overlay will transfer a copy of the software to those without the overlay (GPC-to-GPC transfer via the MM buses).

C. If none of the targeted GPC's contain a G3 archive, then the source of the overlay is the mass memory.

It should be noted that after a "fast" or "move and transfer" transition, the original copy of the G3 overlay will still exist in upper memory.

11.1.5.4 GPC Initialization/Moding to Run and G3 Archive

The following is the logic each GPC performs upon GPC initialization or mode to RUN.

A GPC is brought into an operational state in an orderly fashion called GPC initialization. There are two phases of GPC initialization: IPL and PASS. The second phase is executed immediately after an IPL or when the GPC is moded to STBY/HALT and back to RUN without a fresh IPL. For the G3 archive, if the PASS initialization is the result of an IPL, each GPC will store the destination address table (DAT) in upper memory and the G3 archive status (an MCC status) is zeroed out. The DAT contains the source/ destination addresses for moving the G3 archive overlay from upper memory to lower memory. This table is identical for each GPC.

Also, upon primary GPC initialization (the first GPC moded to RUN), SPEC 1 will exhibit the G3 archive load function enabled and the G3 archive retrieve function disabled. However, the initialization of item 48 enabled does not imply that load function is valid in OPS 0.

11.1.5.5 G3 Archive Discrepancy Reports

The following is an overview of all the discrepancy reports (DR's) that have been written against the G3 archive function. The asterisk (*) denotes that an OPS note/waiver has been written for that particular DR which should be referenced for additional information.

A. *DR 103900 - SPEC 1, ITEM 11 IS REJECTED INCORRECTLY DUE TO NEW G3 ARCHIVE CODE

This DR states that, if the G3 archive load function (item 48, SPEC 1) is enabled, the GPC/LDB source bus (item 11, SPEC 1) is an illegal item entry and vice versa. The DR was written for a scenario when item 48 is enabled in a nonvalid memory configuration; i.e., an unsuccessful G3 archive load. An OPS note/waiver is written for the OI-20 software, and the code will be fixed for OI-21.

B. DR 105704 - G3 ARCHIVE RETRIEVE DOES NOT ALLOW MMU FORCE OVERLAY OF OPS

This DR states that, if the MMU/MMU source bus (item 10, SPEC 1) is selected when a G3 transition occurs with a G3 archive available and G3 is not in lower memory, the G3 overlay is retrieved form the GPC's upper memory instead of being forced from the MMU. A documentation CR will be written to exclude the G3 archive, when the retrieve function (item 49) is enabled, from the requirement allowing the user to force the overlay from the MMU when item 10 is selected.

C. *DR 104314 - THE ARCHIVE RETRIEVE FLAG NOT INITIALIZED TO OFF

This DR states that, if retrieve function (item 49) is enabled when a GPC is moded to STBY and later moded back to RUN as the primary GPC, both the load and retrieve functions will be enabled simultaneously. Requirements state that item 49 should be disabled upon primary GPC initialization. An OPS note/waiver is written for OI-20 software, and the code will be changed for OI-21.

D. DR 104334 - G3 LOAD PENDING FLAG NOT RESET AFTER PRE-POSITION ERROR

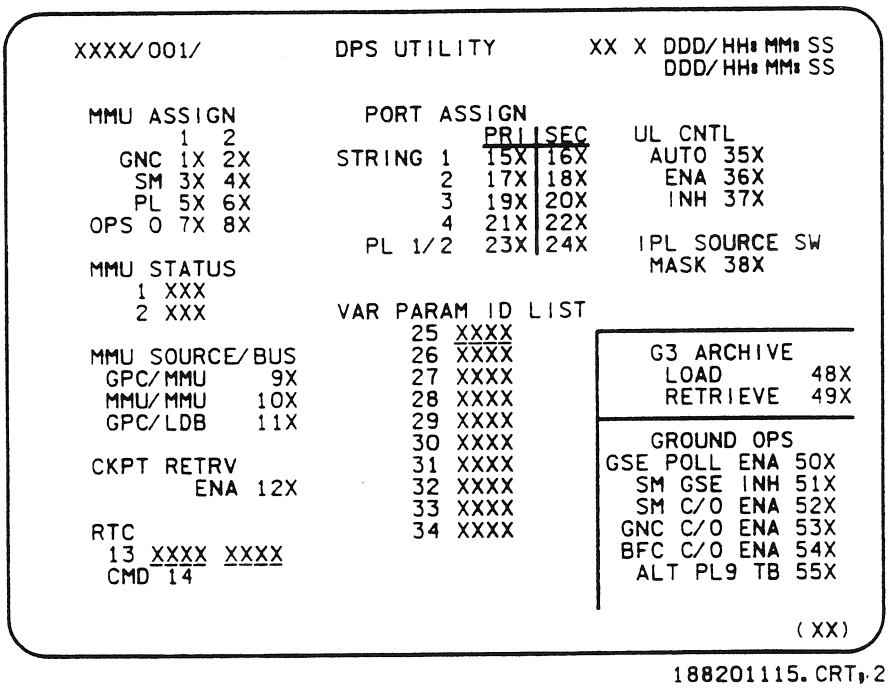
This DR states that, if pre-position failures occur on both MMU's during a G9-to-G1 transition and with the G3 load function (item 48) enabled, the G3 archive load pending flag will not be reset. Thus, upon a subsequent OPS transition, the G3 archive could be loaded into all of the GPC's lower memory overwriting the current contents even though item 48 indicates that the function is disabled. A source fix will be done for OI-20.

E. *DR 104335 - THE ARCHIVE RETRIEVE (ITEM 49) DISABLE RECEIVES "ILLEGAL ENTRY"

This DR states that, if no G3 archive overlay is available in any GPC but the retrieve function (item 49) indicates enabled, an ILLEGAL ENTRY message will be annunciated when an item 49 entry is made to try and disable the function. An OPS note/waiver is written for OI-20, and a code change will be implemented for OI-21 to correct the problem.

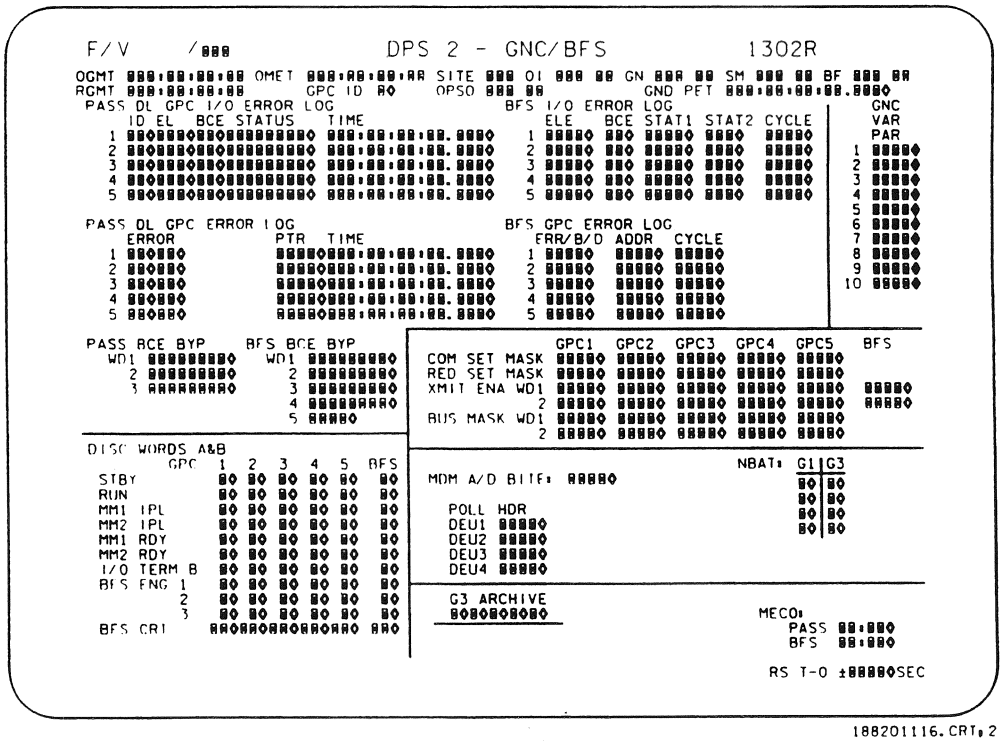
F. *DR 105712 - G3 LOAD PENDING FLAG NOT RESET AFTER ERROR CONDITION IDENTIFIED (OI-21)

This DR states that, upon a G9 to G1 transition with both item 48 (G3 archive load) and item 11 (GPC/LDB) selected on SPEC 1, the G3 load pending flag will remain set. The transition is aborted and an ILLEGAL ENTRY message is annunciated. If the next transition is a G9 to G1 transition using the mass memory, the flag will be reset and no further software problems. However, if the next transition is not a G9-to-G1 transition, extensive software problems could occur (i.e., GPC BITE messages or even an invalid PASS software load). An OPS note/waiver is written for OI-21.



188201115.CRT;2

Figure 11.1.5-1.- SPEC 1, DPS UTILITY display.



188201116.CRT;2

Figure 11.1.5-2.- MSK 1302, DPS 2 display.

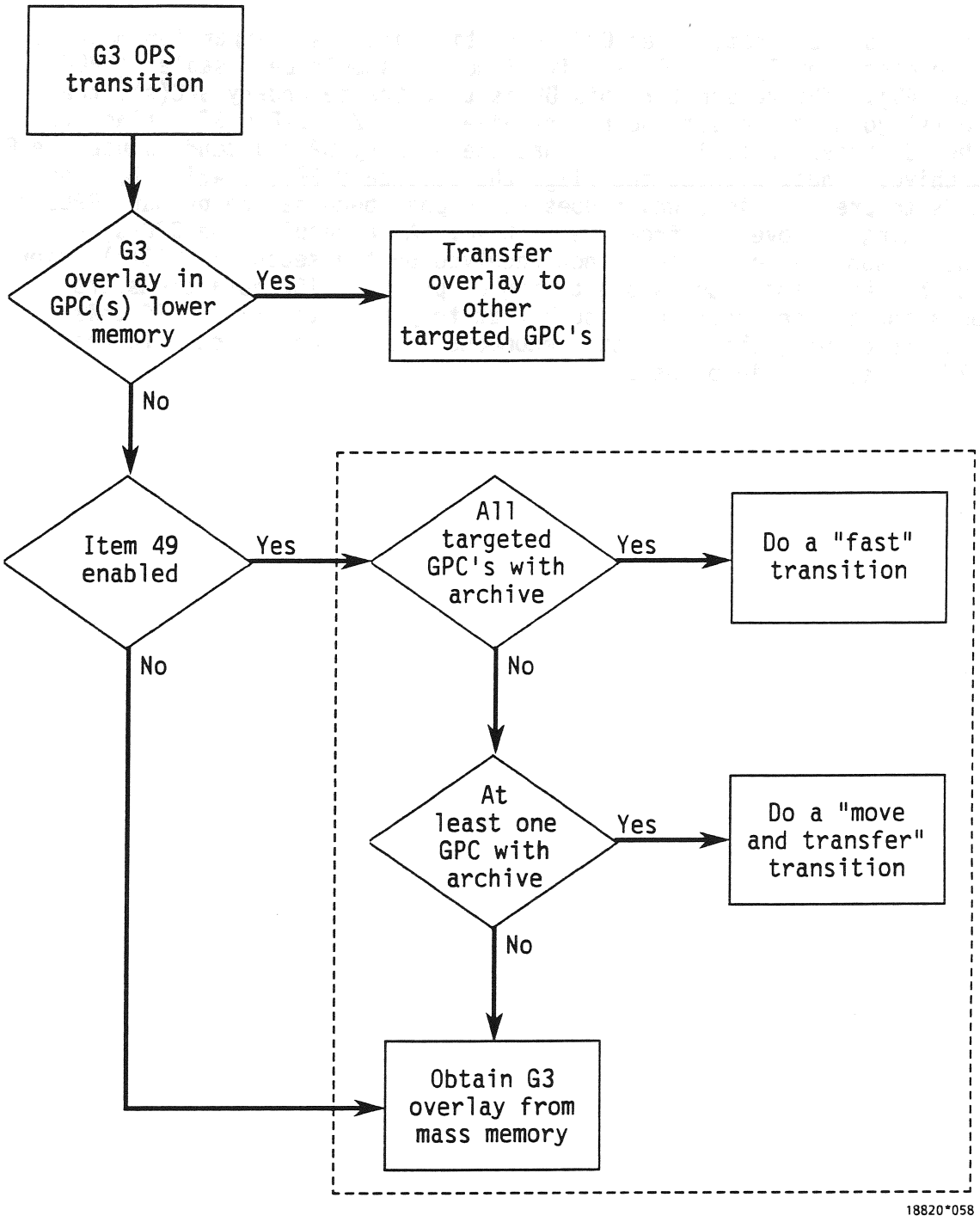


Figure 11.1.5-3.- Flowchart for G3 overlay process.

G. *DR 104860 - GPC FTS AT OPS 3 OVERLAY

This DR states that, if an OPS transition and a set expansion must be done concurrently, the G3 archive function should be disabled (SPEC 1, item 49). The reason for this DR is that the secondary GPC(s), the one(s) joining the set, do not receive the CZZB G3ARCH STAT flag over the ICC buses to tell the GPC that the primary GPCs intend to use the G3 archive. Thus, without the flag, the secondary GPC(s) waits for the MMUs to pre-position, which does not happen because the primary GPCs are obtaining the overlay from the archive. As a result, the GPC(s) FTS. The reason why this flag is not received by the secondary GPC(s) prior to the OPS transition is due to higher priority ICC data transfers delaying the secondary GPCs such that they miss the flag. If this failure occurs, the User Note recommends an OPS mode recall to bring the FTS GPC(s) back into the set.

11.2 USER INTERFACE[1]

The user interface is a permanent resident set of modules designed to support users of flight software. It is that part of the Space Shuttle software which provides the interface between the user or ground and the various application software. The software provides the user and the ground with the capability to monitor and control the software functions. This is accomplished through the various command inputs and support of application control segments through the use of a control segment grammar, which defines the limitation and constraints of the control segments. It also provides the exclusive medium for two-way communication with the crew via CRT displays and keyboard entries, for communication with the ground via the LDB and downlist, and for data exchange between computers via the ICC.

Figure 11.2-1 illustrates the data and processing flow through user interface with inputs from the ICC, LDB, and keyboard interfaces. The chart depicts the major areas in user interface and their relationship to one another in the processing of inputs to present displays on the CRT, to support the applications, and to transfer data to the ground.

All references made in this section are to the Space Shuttle Programs, Operational Detailed Design Specification (DDS), volume II, part 2 - User Interface, unless otherwise specified.

The user interface is divided into three major areas (fig. 11.2-2) which are discussed as follows.

Command input processing provides the flight software with user inputs from the MCDS, messages from the ground through the network signal processor (NSP), and Intercomputer channel (ICC) communications from the other GPC's (referenced document, section 3.1).

Operations control performs initiation, sequencing, and termination of application and system services processing as requested by command inputs and/or application services (referenced document, section 3.2).

Output message processing and coordination provides flight software with the capability of presenting displays and messages to the user through the multifunctional CRT display system (MCDS), of commanding the C&W indicator alarm, the alert light and the alert tone, and of communicating with the ground by way of the downlist and launch data bus (also included is the input processor from the LDB) (referenced document, section 3.3).

A. Command input processing

The command input processing software conducts the communications necessary to obtain inputs from the MCDS, the NSP, and the ICC. In addition to the acquisition of these inputs, the routing of commands and data to target GPC applications software is provided. It has three major subdivisions (fig. 11.2-3) as follows:

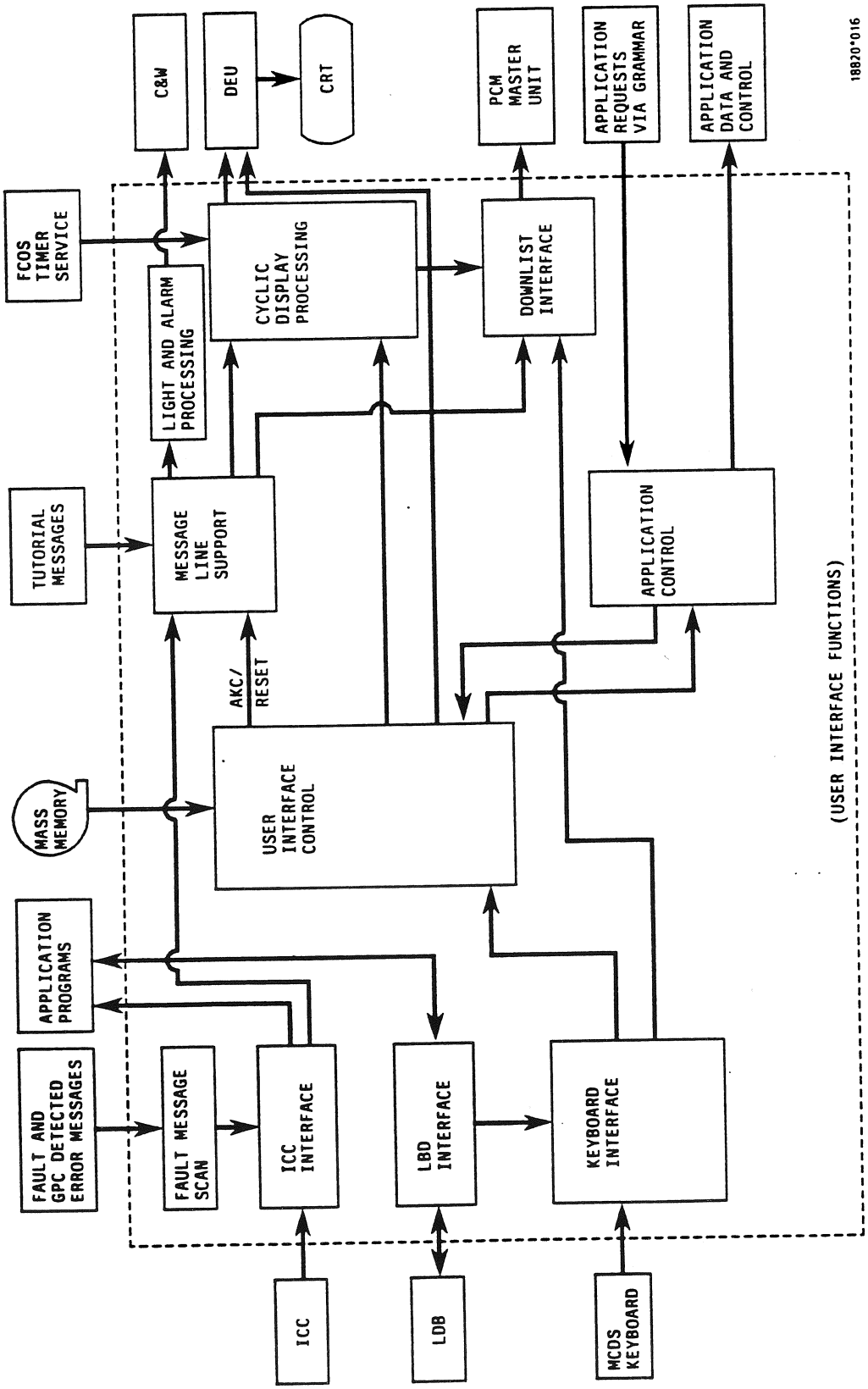


Figure 11.2-1.- User interface processing flow.

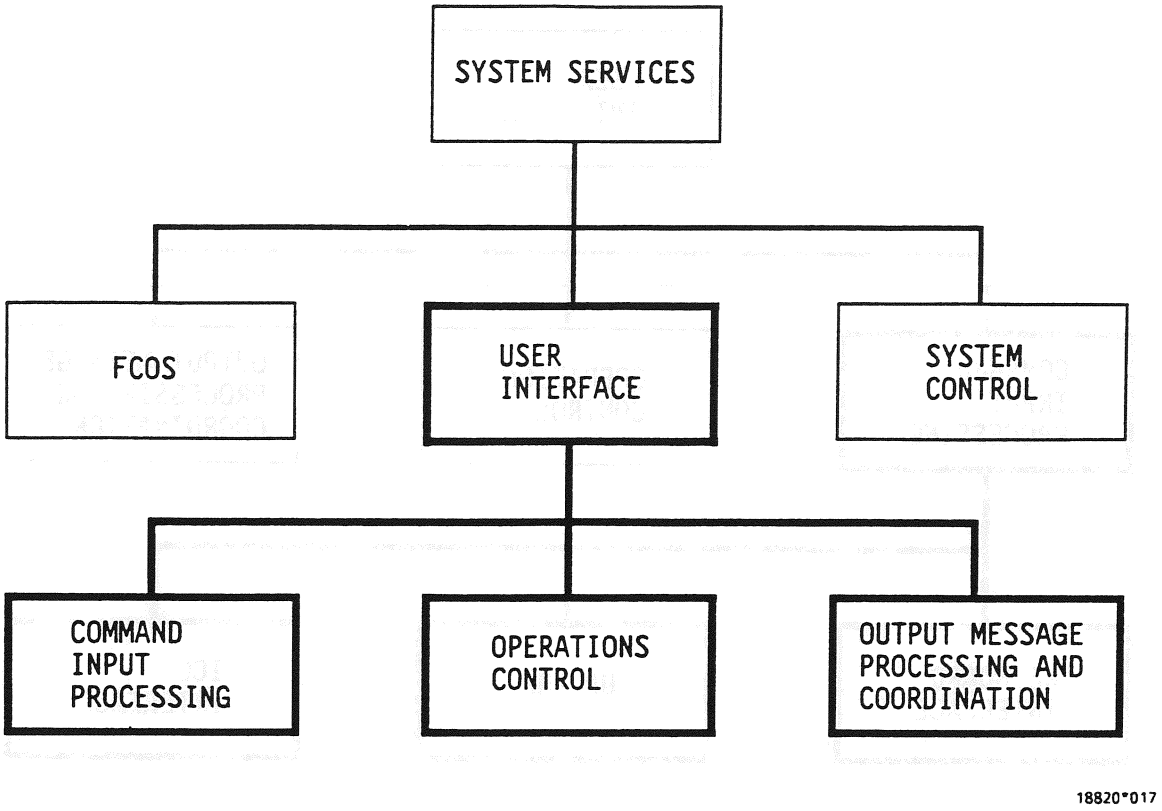


Figure 11.2-2.- Functional overview.

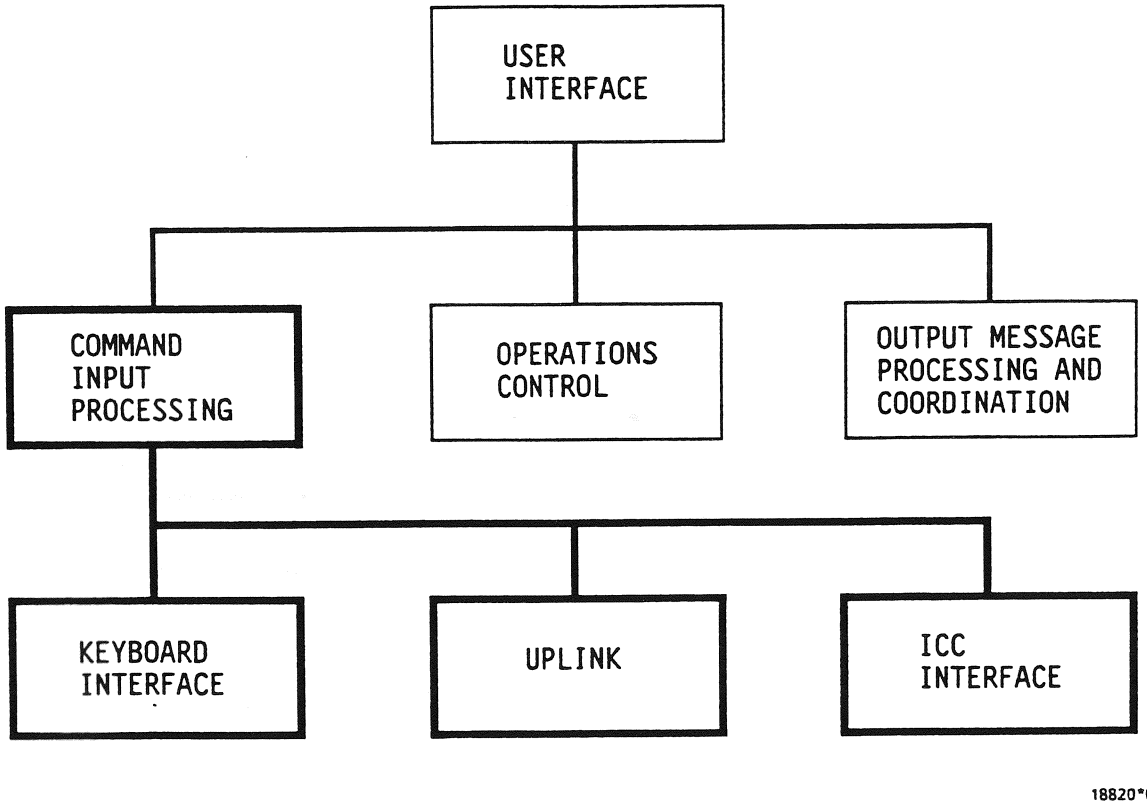


Figure 11.2-3.- Command input processing hierarchy diagram.

1. Keyboard interface software provides input communications with the MCDS and routes the data acquired from it to operations control for further processing and response. The software is divided up as follows:

   MCDS input processor (DMI_MCDS_IN - 3.1.1.1)
   MCDS message processor (DMM_MCDS_PROCESS - 3.1.1.2)

2. The uplink software contains the common user interface software required to support uplinks to the Orbiter via the NSP. The software is divided up as follows:

   NSP message processor (DUP_NSP_MSG_PROC - 3.1.2.1)
   NSP stored program command processor (DSP_SPC_PROCESSOR - 3.1.2.2)
   NSP G MEM processor (DGM_WRT - 3.1.2.3)
   NSP time initiated I/O (DUM_ULK_IO - 3.1.2.4)
   Variable downlist update (DDL_VARIABLE_DOWNLIST - 3.1.2.5)

3. ICC interface provides common support for ICC messages required to be transferred among the common set of GPC's at common sync time. A service is provided to accumulate message between system interface program (SIP) cycles, to merge them into a single message for transfer to all other GPCs in the common set, and to unpack the messages from all GPCs and make them available to the processes for which they are destined. The software is divided up as follows:

   ICC message collector (DIM_ICC_COLLECTOR - 3.1.3.1)

B. Operations control

The operations control software decodes user inputs (from command input processing) and coordinates the user to GPC processing. It handles the orderly termination of application processes, memory overlays, and initiation of new processes. It has two major subdivisions (fig 11.2-4) as follows:

1. User interface control receives user commands in the form of MCDS defined messages from command input processing and application service requests from application control. It coordinates the execution of OPS transition, SPEC, and display functions by MF and DEU. The software is divided up as follows:

   User interface control supervisor (DMC_SUPER - 3.2.1.1)
   Keyboard data processor (DM1_KEYBOARD - 3.2.1.2)
   Application request processor (DM2_APPLICATION - 3.2.1.3)
   Page locator (DM3_DISPLAY - 3.2.1.4)
   DEU background update processor (DM4_DEU_PROC - 3.2.1.5)
   Control segment scheduler (DM5_NEW_CSEG - 3.2.1.6)
   OPS request processor (DM6_OPS_PROC - 3.2.1.7)

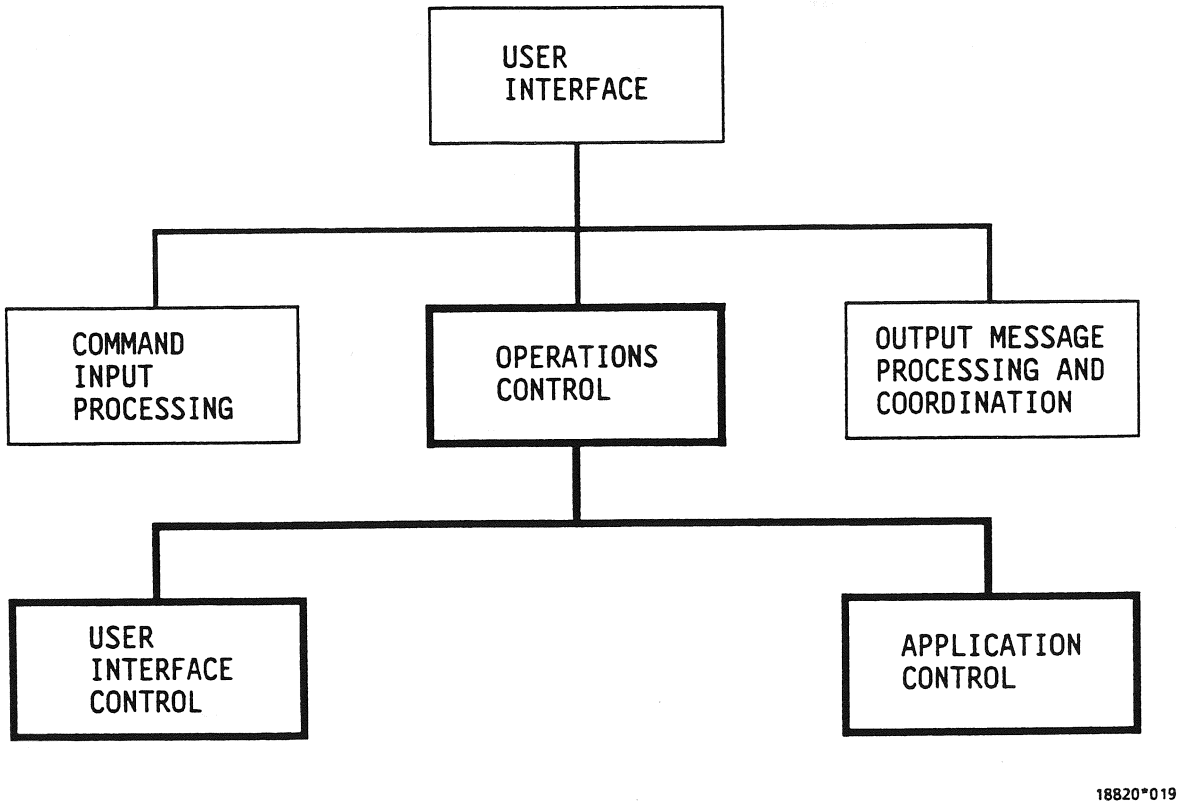


Figure 11.2-4.- Operations control hierarchy diagram.

Request MF termination processor (DM7_REQ_MFTERM - 3.2.1.8)
SPEC request processor (DM8_SPEC_PROC - 3.2.1.9)
Item processor (DM9_ITEM - 3.2.1.10)
Buffer addressability processor (DMN_NEW_LOAD - 3.2.1.11)
Resume DEU processor (DMR_RESUME - 3.2.1.12)
UI control supervisor log function (DMZ_LOG - 3.2.1.13)
Cargo control SPEC processor (DMC_CC_SPEC - 3.2.1.14)
MM rollin display processor (DXR_DMM_ROLLIN - 3.2.1.15)
Control segment termination (DXX_CSEG_TERM - 3.2.1.16)
I/O reset processor (DIR_IORESET - 3.2.1.17)

2. Application control provides the control necessary to maintain user control of the application programs. It also provides display and control services to the application through the control segment. The software is divided up as follows:

   Display presentation and control (DIS_PLAY - 3.2.2.1)
   Application moding and sequencer (DNX_BMS - 3.2.2.2)

C. Output message processing and coordination

The output message processing and coordination software provides flight software with the capability of presenting displays, messages, light, and alarms to the crewmember through the MCDS and of communication with the ground by way of the downlist and launch data bus. It has four major subdivisions (fig 11.2-5) as follows:

1. CRT interfaces generates the format control words (FCWs) which produce the CRT displays. The FCWs are regenerated on a cyclic basis to provide updated display parameters. The CRT interface also supports the error message line and the tutorial message line on the displays. The software is divided up as follows:

   Cyclic display processor (DCICYC - 3.3.1.2)

2. The downlist interface is the means through which the various applications and system services provide data to the ground via the PCMMU. The downlist software collects, buffers, formats, and transfers the data to the PCMMU.

   GPC downlist formatter (DCDDOW - 3.3.2.1)
   Operational data processor (DCD_DG1, DCD_DG2, DCD_DG3, DCD_DS2, DCD_DS8, DCD_DS9, DCD_DG8, DCD_DG9 - 3.3.2.2)

3. The LDB Interface has the responsibility of coordinating messages sent to and from the ground via the launch data bus. It provides the data formatting and buffering required to accomplish this communication. (Reference DDS, volume III, part 3 - Vehicle Utility and Data Flow.)[2] The software is divided up as follows:

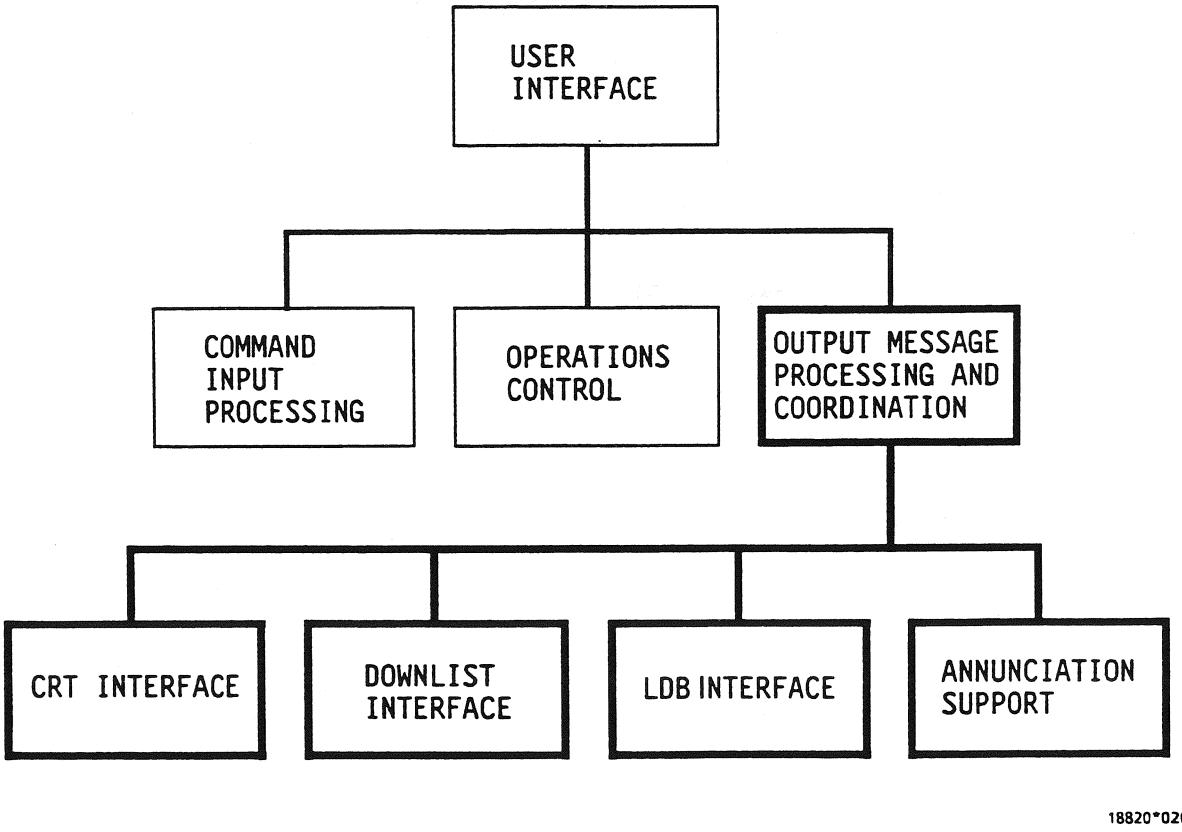


Figure 11.2-5.- Output message processing and coordination hierarchial diagram.

GSE interface processor (DGI_LDB_IO - 3.7.1)
GSE input message processor (DGR_GSE_ROUT - 3.7.1.1)
GSE error processor (DGE_GSE_ERROR - 3.7.1.2)
GSE output message coordinator (DGO_GSE_COORD - 3.7.1.3)

4. The annunciation support software provides the interface for flight software to drive the C&W indicator alarm, the alert light, and the alert tone. It also supports the fault summary page display on the MCDS which contains a history of software-detected fault messages. The software is divided as follows:

   Error message line support (DMT_ERR_MSG - 3.3.4.1)
   Annunciation macro interface (DMA_MAC - 3.3.4.2)
   Light and alarm processor (DLA_LIGHT_ALARM_PROC - 3.3.4.3)

References

1. Space Shuttle Programs operational Detailed Design Specification (DDS), Volume II, part 2 - User Interface, OI-8D, 9-1-89.

2. Space Shuttle Programs Operational Detailed Design Specification (DDS), Volume III, part 3 - Vehicle Utility and Data Flow, OI-8D, 9-1-89.

11.2.1 Uplink Software

11.2.1.1 Introduction

This brief will describe the common user interface software required to support uplinks to the orbiter via the NSP. Uplinks described in this brief are limited to the DF22 OL loads. It will describe the onboard processing for the following uplink loads: DEU equivalents, GPC memory read/writes, MMU patches, table maintenance block update (TMBU), and variable parameters downlist. Processing of the entry memory overlay will be added at a later date.

The user interface software responsible for the command uplink consists of five major processes, which in turn call other processes.

A. NSP_Message_Processor (DUP_NSP_MSG_PROC) provides the logic necessary to control the uplink process. It is responsible for the decoding and verification of input commands from the ground system via the uplink software and the subsequent execution of those commands. DUP_NSP_MSG_PROC is scheduled at GPC initialization (Standby to Run, including IPL) by AIB_GPC_LOCATOR. DUP_NSP_MSG_PROC then waits for either the process event from AIE_SIP or the cancel event which is set during GPC reconfiguration by ARC_GPC_RECONFIG. After RS formation, ARC_GPC_RECONFIG reschedules DUP_NSP. If the process event is set, DUP_NSP will process NSP data read from the current NSP. NSP read is performed by all GPC's in the redundant set with the commander of the bus which the current NSP is on. The GPC's will ICC NSP data read every 160 msec.

B. NSP_Stored_Program_Command_Processor (DSP_SPC_PROCESSOR) is responsible for controlling the buffering and execution of NSP stored program command (SPD's).

C. NSP_G_MEM_Processor (DGM_WRT) determines the formats for G_MEM contiguous and scatter commands and writes into main memory.

D. NSP_Time_Initiated_IO (DUM_ULK_IO) provides the mechanism to perform NSP's cyclic I/O without any process being dispatched.

E. Variable_Downlist_Update_Processor (DDL_VARIABLE_DOWNLIST) handles requests to update variable parameters for downlist received by the DUP_NSP_MSG_PROC.

11.2.1.2 Onboard Command Types

Two types of NSP commands may be uplinked to the orbiter, single-stage or two-stage. For the most part, DPS (DF22) uplink command loads are two-stage commands with the exception of the OPS 3 overlay uplink, which is a unique case and will be addressed later. A description of the DPS two-stage

command processing will be addressed by this brief. For details on single-stage commands, please refer to JSC document JSC-18611 (INCO/COMM Systems Brief).

The high-order bit of the Op code set is used to determine whether a command is two stage or single stage. Two-stage commands have the high-order bit of the Op code set to zero; whereas, single-stage commands have the high-order bit of the Op code set to one.

Two-stage commands are placed in a two-stage buffer and downlisted so the ground can verify the command prior to execution. A two-stage execute command must be uplinked to execute the two-stage command. Two-stage commands with Op codes of a value less then six are also executed by uplink software; this includes command loads GMEM contiguous, GMEM scatter, and DEU equivalents. All other two-stage commands are passed to the application uplink processor (command loads include GMEM contiguous, GMEM scatter, DEU equivalents, GPC memory read/writes, TMBU, variable parameter, and memory overlay uplink). Table 11.2.1-I lists the DPS commands and Op codes.

TABLE 11.2.1-I.- DPS CMD AND OP CODE REFERENCE TABLE

Command type or uplink message	Op code			48-bit command words
	Binary	Hex	Octal	
ILLEGAL	0000000	00	000	
GMEM CONTIGUOUS	0000001	01	001	2-33
GMEM SCATTER	0000010	02	002	2-33
EQUIVALENT DEU	0000101	05	005	6
GPC MM READ	0000110	06	006	2
GPC MM MERGE	0000110	06	006	4-33
GPC MM WRITE	0000110	06	006	2
TABLE MAINTENANCE BLOCK UPDATE	0100001	21	041	4-3I
VARIABLE DOWNLIST SELECT	0I10111	37	067	10
TWO-STAGE BUFFER CLEAR	1000001	41	101	1
SPC BUFFER CLEAR	1000010	42	102	1
TWO-STAGE BUFFER EXECUTE	1000011	43	103	1
MEMORY OVERLAY UPLINK	1001000	48	110	2-33

11.2.1.3 Two-Stage Command Processing

Once a two-stage command is initiated from the MCC, the Mission Operations computer will transmit thirty-three 48-bit uplink command words to the orbiter. Each command word (fig. 11.2.1-1) will contain the following information.

The command header word (first 16 bits in a command) will always contain the vehicle address, major function, Op code, first command word, and last command word. The remaining 32 bits will contain the data portion of the command.

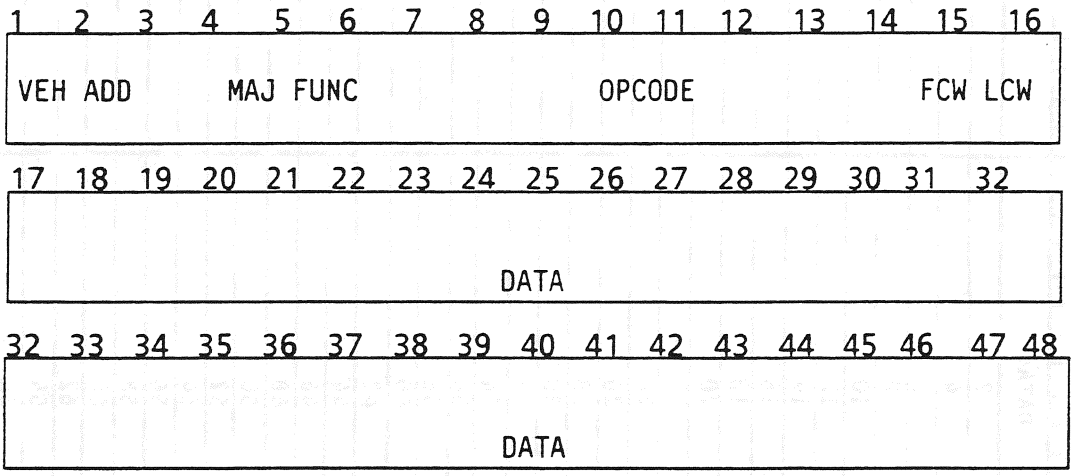


Figure 11.2.1-1.- Command word structure.

The module DUP_NSP_MSG_PROC is executed to process NSP data read from the current NSP. The NSP read is performed by all GPC's in redundant set with the commander of the bus which the current NSP is on (MDM FF01 for NSP1, MDM FF03 for NSP2). The inputs to the module DUP_NSP include 30 halfwords (ten 48-bit command words) of uplink data. In addition to the uplink data, the NSP validity word, the NSP discretes, and power discretes are also read off the NSP but are not used by DUP_NSP_MSG_PROC as inputs; these words are used for error checking. These GPC's will ICC NSP data every major cycle (160 msec).

The onboard two-stage buffer (fig. 11.2.1-2) will store up to sixty-seven 16-bit words. All 48-bits of command word one are stored, and only bits 17 through 48 of successive words are stored. The first 16 bits of all command words are identical, with the exception of bits 15-16.

If the first bit of the Op code is set to zero, then two-stage commands are processed. DUP_NSP will perform error processing up front prior to executing the command load. When NSP data is processed, two validity words (GPC (bits 11-16) and NSP (bits 11-16) validity) are generated to indicate error conditions detected by the software and are downlisted to the ground.

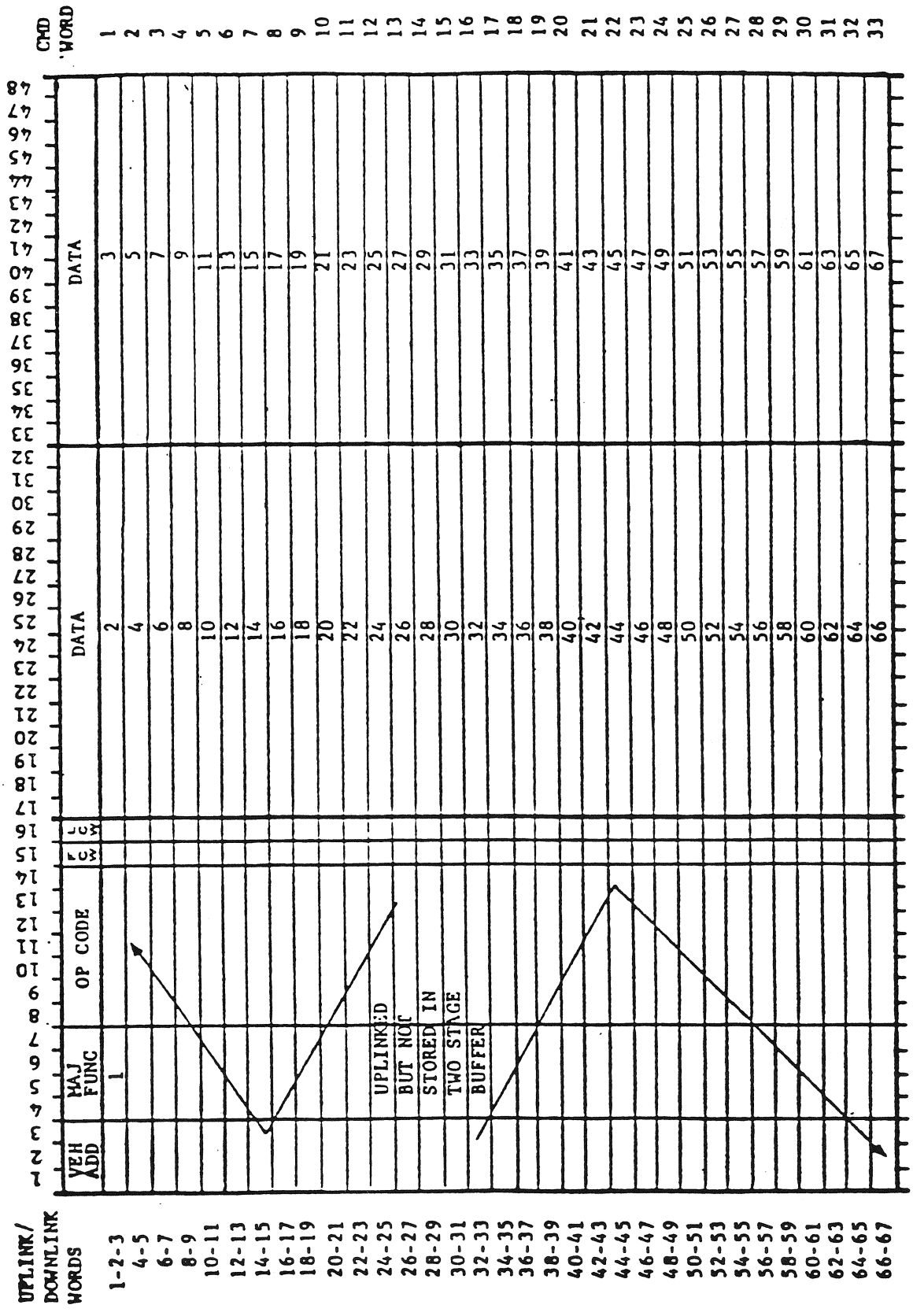


Figure 11.2.1-2.- Orbiter two-stage command buffer.

The GPC validity word (fig. 11.2.1-3) is generated in response to each uplink command and indicates the command word (1 to 10) in error. Bits 1-10 consist of validity indicators, and bits 11-16 consist of error indicators for the following error conditions: GPC/MF check, uplink process skip, SPC buffer full, illegal Op code/MC, and two-stage type error. In addition, the uplink software uses the spare portion of the NSP validity word (bits 11-16) for additional error indicators.

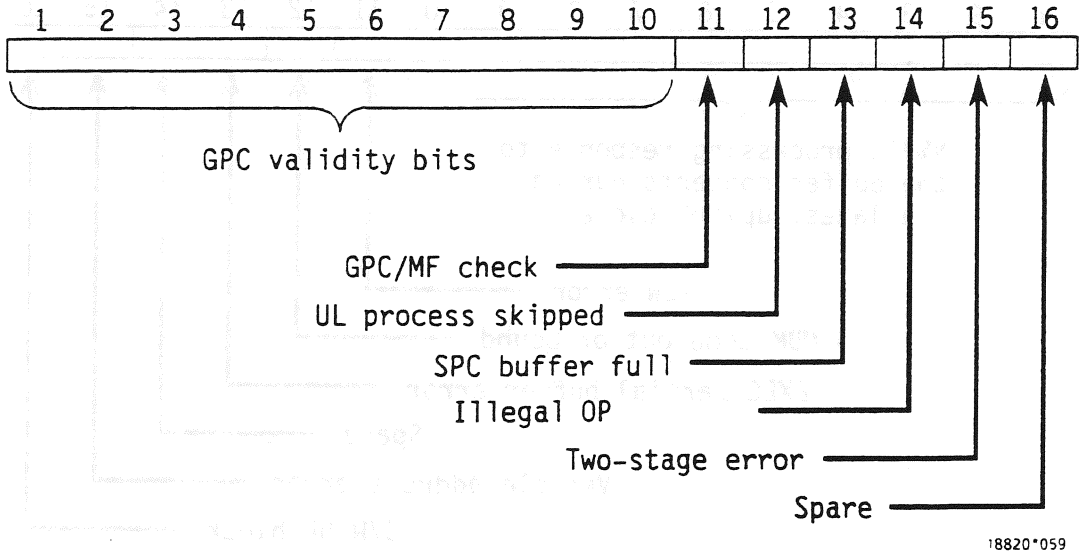


Figure 11.2.1-3.- GPC validity word.

If the command is to be executed by the application and a major function was not specified, the GPC/MF bit is set in the GPC validity word. If the reconfiguration in progress indicator is set on or the Op code/MC bit is off (Op code ILLEGAL for this memory configuration), then set the Op code/MC bit. If the first word of the two-stage is not zero and the command Op code does not match the buffer Op code, then the two-stage type error bit is set. The software will determine the first command word in a load. If first the command word and the first word of the buffer are not zero, the FW/LW bit is set. If the two-stage buffer is full, the two-stage type error bit will be set.

(The UL process skip bit is set.) DUP_NSP will determine the availability of the two-stage buffer. If the application skip flag is not zero, then the two-stage buffer may not be available. If the application event is set or the buffer has data in it, set the local skip flag to one; otherwise, the application skip flag is zeroed.

The NSP validity word (fig. 11.2.1-4) is part of the data read from the NSP. The first 10 bits are hardware generated and indicate those command words which are valid; the remaining bits indicate first/last word error, bad MDM code, partial buffer execute, vehicle address error, and software uplink block. Note the validity bits reflect each read of the NSP and are dynamically updated while the error condition indicators only reflect the error of the last uplink command.

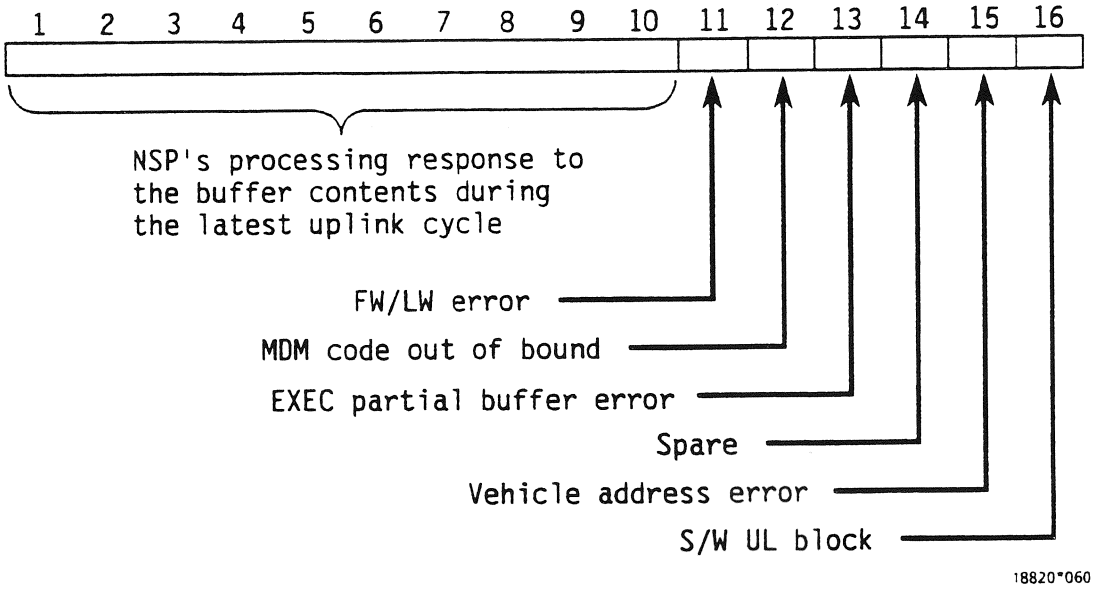


Figure 11.2.1-4.- NSP validity word.

Upon receipt of the single-stage command, TWO STAGE BUFFER EXECUTE, the module DUP_NSP_MSG_PROC will call DUP_2_STAGE_EXEC. DUP_2_STAGE_EXEC will process the commands with Op codes five or less, and notify the proper application for processing of other Op codes.

11.2.1.4 GPC Main Memory Read/Writes

If Op code equal to one or two, then DUP_2_STAGE_EXEC will schedule DGM_WRT to process the command and to clear the buffer. GMEM contiguous has an Op code of 1; GMEM scatter has an Op code of 2.

DGM_WRT processor will format data for uplink GPC memory writes. The actual GMEM writes are performed by issuing the FCOS program modification macros (PMODLIST and PGMMOD).

The data used by DGM_WRT is stored in the uplink two-stage buffer upon entry. The data is shipped to DGM_WRT in the following format:

Word 1	Bits	Purpose
1	8-14	Uplink two-stage Op code BIN '0000001' = GMEM contiguous write BIN '0000010' = GMEM scatter write
2	4-5	Load/set/reset indicator BIN '00' = Invalid BIN '01' = Set bits in memory BIN '10' = Reset bits in memory BIN '11' = Load data into memory
2	7	Protect/unprotect indicator BIN '0' = Unprotected location(s) BIN '1' = Protected locations
2	8-14	Number of words (contiguous) or number of address/data pairs (scatter)
2	15-16	Most significant bits of the address(es)
3	1-16	Starting address (contiguous)
4	1-16	Data field (contiguous) Address/data pairs (scatter), one halfword for each address or data item

The software will pick up the most significant bits of the data address and the load/set/reset indicator. If the request is valid, then processing continues; otherwise, the program terminates.

If the write is for protected locations, set up the SVC parameter list appropriately (i.e, set bits, reset bits) or load data into memory.

For an unprotected write, set up the parameter list in the same way by entering another case structure.

If the write is contiguous, set up the remainder of the SVC parameter list. Load the starting address into the parameter list. If the request is a load, then pick up the number of words and move the data to the data buffer; otherwise, set up for a length of one and move the data to the set/reset mask. Issue the SVC to perform the actual write.

Determine a GMEM scatter write, determine the number of address/data pairs, and set the write length of one. Starting with word 4 in the two-stage buffer, enter a loop to process one address/data pair at a time. Set up the address and move the data to the proper location. Issue the SVC to perform the write.

After the two-stage processing is complete, the two-stage buffer is cleared.

11.2.1.5 DEU Equivalents

If the Op code is equal to five, then the DEU equivalent will be processed. DUP_2_STAGE_EXEC will make sure the number of keystrokes is between 1 and 30 and then determine the DEU number. If the DEU number is from one to four, then the DEU equivalent message is processed.

If no message is waiting to be executed, copy the equivalent bit, copy the DEU number, clear the two-stage buffer, and set the DEU equivalent flag for polling.

11.2.1.6 Variable Downlist

A variable downlist select has an Op code of 55. DUP_NSP_MSG_PROC will be called to process the variable downlist. The DDL_VARIABLE_DOWNLIST processor module is responsible for updating the variable parameter address list based on inputs from the uplink two-stage buffer.

The uplink two-stage buffer contains the data necessary to redefine the VP selections for downlist. The data in the buffer is organized into groups of two halfwords. The first contains the number (1 to 10) of the downlist parameter to be updated; the other contains the address to be downlisted (i.e., '0000' to 'FFFF').

DDL-VARIABLE_DOWNLIST updates the appropriate location in the DPS UTILITY SPEC (CDJV_WORK_AREA) and the downlist address for the downlist data collector (CDWV_CUR_VAR_DUMP).

If the command is to be processed by the application, determine the MF/GPC of the buffer command. If it is the same as that of the two-stage execute (to prevent application uplink execution in ops 0-00), then decrement by six to get a major function index. If the major function index is valid, set the application event for the MF and set the application skip flag to the major function index value.

If the MF/GPC settings do not match, set the GPC/MF bit in the validity word.

11.2.1.7 References

1. NSTS Downlist/Uplink Software Requirements, SS-P-0002-1400.

2. NSTS Space Shuttle Telemetry and Command Data Characteristics Handbook-Command Uplink Volume II, NSTS 08118, vol. 11, rev E.

3. INCO/COMM Systems Brief, JSC-18611, .

4. INCO section members - Glenn Binkley and Joe Gibbs.

5. Flight software microfiche ID OF020100.

11.2.2 Downlink Software

The purpose of this brief is to explain the downlist processing for both the PASS and BFS systems. It is also intended to explain how the downlists interrelate with the overall Shuttle data flow.

11.2.2.1 Overview

The purpose of the downlist processor is to provide data to ground controllers. This data can be used for monitoring the status of a system or a procedure. Alternatively, the data could be given to vendor engineers for resolution of a systems problem. There exist two different data collection systems onboard the Orbiter, the operational instrumentation (OI) system and the downlist system. The rules for each are different, this is a downlist brief not an OI brief. For the OI rules see the Instrumentation and Communication Systems Briefs 22 and 48; then talk to an instrumentation engineer.

The downlist software is of very simple nature. All it does is fetch the data located in a memory location and throughput this data to the input/output (I/O) systems software. (The downlister is considered a part of systems software; however, what it downlists can be considered an application.) The I/O software conveys this data to the communications system via the instrumentation pulse code modulation data (IP) bus, which is how the pulse code modulation master unit (PCMMU) gets the downlist data. The PCMMU then multiplexes the downlist data with OI data and payload data interleaver (PDI) or Spacelab data, forms an operational data frame, and ships this frame to the NSP's.

NSP's multiplex the operational data frame and two digital voice channels to form an operational downlink. This operational downlink can be recorded on board and subsequently downlinked to ground to cover LOS periods.

When a site receives the downlink, it formats the data frame into a NASCOM block format. The site then sends the data over NASCOM communication links to Mission Control Center (MCC). At MCC the voice information is stripped from the data, the data is shipped to a telemetry preprocessor computer (TPC) which in turn ships the data (usually a subset of all the data that is downlinked) to the MOC and the analog event system (AES) which conditions signals so they can be put on strip charts. Figure 11.2.2-1 shows a summary of this flow.

11.2.2.2 Downlist Processor

The downlist software is restricted to obtaining data from sectors 0 and 1 of memory (addresses 0000 to FFFF). Thus, those applications which have downlist requirements must see to it that those parameters downlisted are in sectors 0 and 1. The downlist processor is table driven. The table consists of a series of addresses for the parameters that will be downlisted. The tables are both memory configuration and downlist frame count dependent.

The processor is activated 25 times a second by the System_Interface_ Processor (AIE_SIP) via a call for DCD_DOW. Each time it is activated it creates a frame of data according to a table which is frame count and memory configuration dependent. The term memory configuration is used advisedly (e.g., GNC 201 and 202 have the same downlist table). One hundred Hz is the highest frequency that a given parameter is downlisted. To do this, four time samples of a parameter are put into each downlist frame. The frames can contain up to 128 sixteen-bit words of which seven words are overhead. Figure 11.2.2-2 gives the frame size for each memory configuration. The maximum bandwidth that the downlist is presently designed to support is 51.2 kbps. There is something called low data rate (LDR) which as a rule is half of the correspondent high rate. There is no technical reason why LDR must be one-half the high data rate (HDR). It is just easier for the telemetry format load (TFL) compiler to achieve a solution when the ratio is one half. Thus, LDR for a 51.2 kbps downlist is 25.6 kbps. If for some reason we are downlinking the low operational downlink bandwidth, the HDR downlist data can be dumped off the OPS recorders at a later time (see note below for a caveat).

Another thing to remember is that LDR means half the data not a reduction in rate (e.g.,if a 5 Hz parameter is in the LDR, it will come down at 5 Hz not 1 Hz). There is no LDR downlist processor per sec. The PCMMU has an low rate formatter which picks up half of the downlist toggle buffer (TB) thus creating the LDR. However, there is a special situation in which a HDR TFL allows more bandwidth for payloads than normal. In this case the general purpose computers (GPC's) detect that one of these special HDR TFL's is being loaded into the PCMMU by systems management (SM) and change formats and put out a reduced amount of data to the PCMMU. This is a one-shot occurrence out of the SM machine so, if the other common set machines did not hear it, their downlist format would not change and the downlist will be missing. The only way to recover is to take GPC's out of the common set and put them back in the common set. The LDR indicator (CZ28HDRLDRIND) is part of ICC initialization. You need to reload the TFL if SM is moved to another GPC, since the TFL number loaded is not saved on the checkpoint. If we must go to the compatible LDR format in this situation, the present designs reduce the OI content rather than reduce the downlist bandwidth further.

NOTE

There are several HDR telemetry formats which contain reduced downlist to allow payloads to have a large portion of the 128 kbps bandwidth. In this case reduced downlist is being sent to the recorders and no HDR exists. These are mission dependent; talk to the instrumentation engineer for each mission to determine if this situation will occur on your mission.

11.2.2.3 Formats[1]

Table 11.2.2-I presents all the known formats. It gives the downlist format number, the bandwidth allowed in the operational downlink, the number of words per frame, and finally the memory configuration where each format is available.

11.2.2.4 Downlist Controls

There are three kinds of controls for the downlist. There is a sequence of item numbers which can be used to define the frame length of a dump downlist, the start and stop address of the area to be dumped, and finally execute the dump. (See System Briefs 11.1.3, HISAM Dump, and 11.2.3, Software Dump.) There is also a control which defines the GPC that will talk to TB 1 in the PCMMU. The permissible values to enter with item 44 are 1(2,3,4,5). This is the only validity check made by the computer upon execution of the item number. The software module that starts executing when this is done is called GPC__MEMORY__ITEM. It alters the GPC ID for TB 1 and calls the intercomputer communication (ICC) routines (GPC__ICC__ Collector) which places the identity of the new TB 1 downlister into the set of data that is ICC'd. This allows the other computers in the common set to know who the TB 1 user is and thus know that they should not send data to TB 1 of the PCMMU. Finally, there is a control called OPS 0 ENA which allows the non-prime OPS 0 GPC to downlist through TB 4 (and be sure to use a TFL that talks to TB 4, see JSC 10784 for TFL definitions). This option is used if a dump of an OPS 0 machine is required. The display used for these controls is given in figure 11.2.2-2.

11.2.2.5 Variable Parameters

There is a special feature in the downlist structure called variable parameters. There are 10 general slots which can be used to downlist the contents of memory locations not specified in the hard coding of the formats. This was necessary because it is not possible to identify all requirements before flights. (Much was learned during the first four Shuttle missions, and a number of changes to the downlist were requested. Also the downlist update cycle was so long that some mechanism was needed to cover flights that occurred before new requirements could be implemented.) Another reason for variable parameters is that some parameters can only be justified for narrow time windows. To save downlist bandwidth and still see highly desirable data, the variable downlist was invented. The downlist works on a pointer principle. When you change the contents of the pointer, the parameter that will be downlisted is changed. There is an OP code (command) that allows this to be done from the ground, or the crew can do this via items 25 through 34 on the DPS UTILITY SPEC 001 (fig. 11.2.2-3).

11.2.2.6 Downlister

Figure 11.2.2-4 presents a flow chart (or decision chart) which shows how the downlisting computer(s) is chosen and which PCMMU TB it(they) talks to. The basic elements used are GPC ID, memory configuration, and state of item 44. The backup flight software (BFS) computer always talks to TB 5. During ascent and entry the lowest numbered primary avionics system software (PASS) GPC talks to TB 1. On orbit the nominal situation is for the lowest numbered GNC machine to talk to TB 1 and and the SM machine to talk to TB 2. However, the GNC downlister can be changed by the use of item 44. As an exercise for the reader, go through the flow after an Item 44 + 4 exec is done on SPEC 000, assuming a nominal orbit GPC configuration (GPC 1, 2 = GNC 2, GPC 3 = FDG 3, GPC 4 = SM, and GPC 5 = BFS). Item 44 is active in all memory configurations.

11.2.2.7 BFS Downlist[2]

The BFS downlist like the PASS is table-driven. Unlike the PASS, it also uses bit-packing to increase the efficiency of bandwidth use. It was early recognized that, because of severe bandwidth restrictions on the BFS, something had to be done to fit the requirements into the allocated 12.8 kbps. The PASS system carries a 40 percent overhead in downlisting discretes because it does not repack bits; thus, the BFS does bit-packing. The BFS has no variable parameter capability but accomplishes the same function (covers for unforeseen requirements) with mission-dependent word slots. This was primarily for payloads, but they have been used for Orbiter systems data also.

The BFS has four software programs: OPS 1, 6, 0, and 3. A peculiarity of the BFS is the way formats are selected for OPS 0. The two formats we have for the BFS are 12 (in OPS 1 and OPS 6) and 13 (OPS 3).

OPS 0 may have either format 12 or 13 depending on circumstances. If OPS 0 was started from OPS 1 then OPS 0 uses format 12. It will use format 13 if the transition was OPS 3 to OPS 0. The BFS also will load the low data rate TFL (entry) into the PCMMU given a PCMMU switch. This capability needs to be enabled (item 29 on the BFS MEMORY display), and the BFS has to be up and running.

11.2.2.8 References

1. CPDS, vol. 1, book 4, Downlist/Uplink Requirements, SS-P-0002-140N.

2. Backup Flight System Program Requirements Document, MG038100.

3. Space Shuttle Program Orbiter Avionics Software Operational (OPS) Detailed Design Specifications (DDS), NAS 9-14444.

TABLE 11.2.2-I.- DOWNLIST FORMATS

FMT ID	128 kbps		*64 kbps		POST IPL OPS 0-00	MC1 GNC 1,6	MC2 GNC 2	MC3 GNC 3	MC4 SM2	MC5 SM4	MC6 PL 9	MC8 GNC 8	MC9 GNC 9	BFS OPS 1,6	BFS OPS 3	Name
	Size kbps	FR LEN	Size kbps	FR LEN												
12	12.8	32	6.4	16										X		BFS ASCENT
13	12.8	32	6.4	16											X	BFS ENTRY
20	12.8	32	6.4	16	X											POST IPL/OPS 0-00
21	51.2	128	25.6	64		X										GN&C ASCENT/ABORT
22	44.8	112	22.4	56			X									GN&C ON-ORBIT HDR (eff. STS-2)
23	51.2	128	25.6	64				X								GN&C ENTRY
24	19.2	48	9.6	24					X							SM/RMS/PL (eff. STS-2)
24	(1)	(1)	(1)	(1)					X							SM/RMS/PL HDR (eff. STS-5)
25	(1)	(1)	(1)	(1)						X						SM/RMS/PL HDR (eff. STS-5)
26	19.2	48	19.2	48					X							SM CHECKOUT
32	44.8	112	22.4	56								X				GN&C 8 ORBIT CHECKOUT HDR (eff. STS-2)
42	51.2	128											X			INIT AND CHECKOUT GN&C 9
44	51.2	128											X			GN&C 9 PRECOUNT
46	51.2	128											X			FRT - GENERAL
48	12.8	32	6.4	16							X					MM UTILITY PL 9

(1) Format size and frame length are mission-dependent (definition will be in SD78-SH-0044 Payload Carrier Mgmt FSSR)

* The PCMMU subsets the downlist output by the GPC into this downlist window size for the 64 kbps downlink format.

TABLE 11.2.2-I.- Continued

FMT ID	128 kbps		*64 kbps		POST IPL OPS 0-00	MC1 GNC 1,6	MC2 GNC 2	MC3 GNC 3	MC4 SM2	MC5 SM4	MC6 PL 9	MC8 GNC 8	MC9 GNC 9	BFS OPS 1,6	BFS OPS 3	Name
	Size kbps	FR LEN	Size kbps	FR LEN												
52	51.2	128									X					MM UTILITY PL 9
53	51.2	128											X			GN&C 9 FCS CHECKOUT
60	51.2	128											X			GN&C 9 PAYLOAD CHECKOUT (eff. REL 19)
62	22.4	56	11.2	28			X									GN&C ON-ORBIT LDR (eff. REL 20)****
64	(1)	(1)	(1)	(1)					X							SM/RMS/PL LDR (eff. REL 20)****
65	(1)	(1)	(1)	(1)						X						SM/RMS/PL LDR (eff. REL 20)****
72	22.4	56	11.2	28								X				GN&C 8 ORBIT CHECKOUT LDR (eff. REL 20)****
90			**	**	X	X	X	X	X	X	X	X	X	X	X	MAIN MEMORY DUMP
91			**	**					X	X	X					MASS MEMORY DUMP
92			**	**							X					SPACELAB COMPUTER MEMORY DUMP
93	44.8	112			X	X	X	X	X	X	X	X	X	X	X	HISAM DUMP (eff. REL 18)***

(1) Format size and frame length are mission-dependent (definition will be in SD78-SH-0044 Payload Carrier Mgmt FSSR) *

* The PCMMU subsets the downlist output by the GPC into this downlist window size for the 64 kbps downlink format.

** Main memory, mass memory, and Spacelab computer memory dumps take on the window size of the previous downlist format. An item exists that defines the amount of dump data contained in the window. The remainder is filled with zeros.

*** HISAM dump - refer to CPDS SS-P-0002-170, paragraph 4.6.6.2.3, for the definition of the HISAM dump format.

**** The 128 kbps output of these formats shall be loaded identically to the 64 kbps output of the corresponding HDR format. Refer to paragraph 3.5-C.

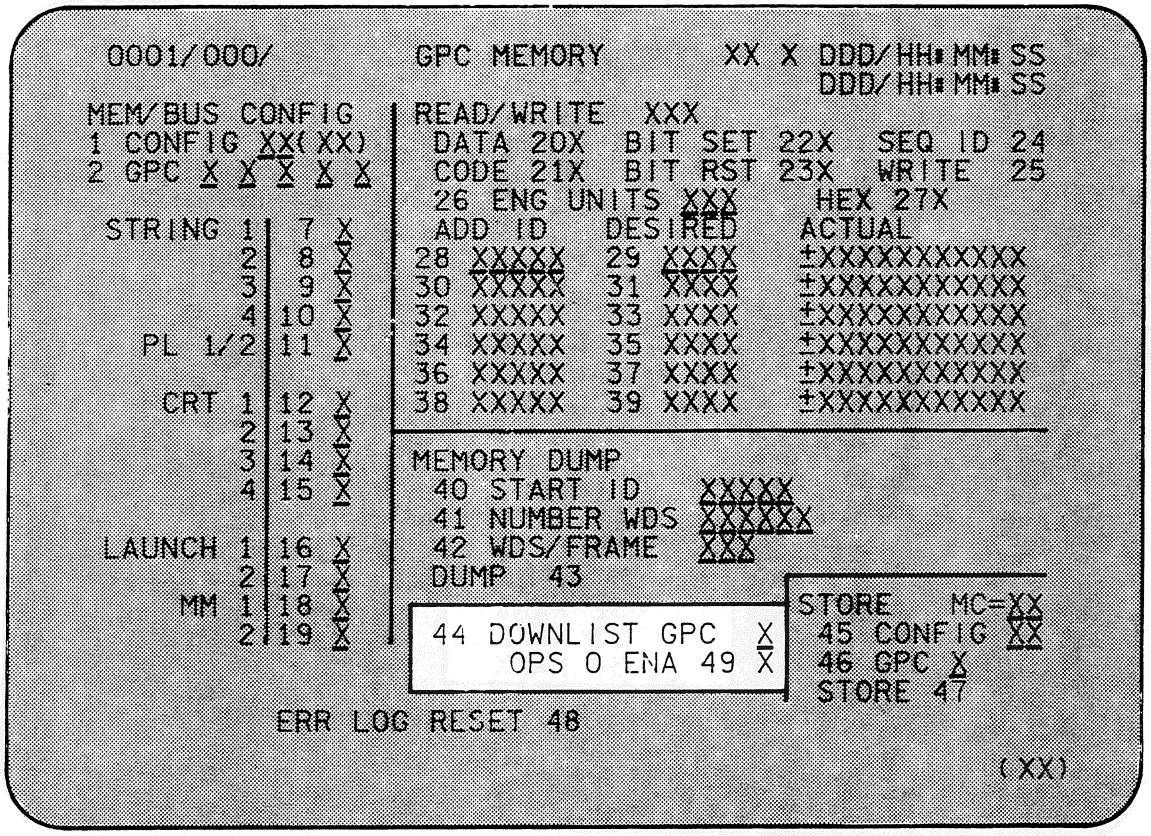
TABLE 11.2.2-I.- Concluded

FMT ID	128 kbps		*64 kbps		POST IPL OPS 0-00	MC1 GNC 1,6	MC2 GNC 2	MC3 GNC 3	MC4 SM2	MC5 SM4	MC6 PL 9	MC8 GNC 8	MC9 GNC 9	BFS OPS 1,6	BFS OPS 3	Name
	Size kbps	FR LEN	Size kbps	FR LEN												
93			**	**	X	X	X	X	X	X	X	X	X	X	X	MAIN MEMORY DUMP (eff. REL 21)
97	51.2	128											X			PCMMU CHECKOUT-ADDRESS 00000 → 052525
98	51.2	128											X			PCMMU CHECKOUT-052525 (OCTA L) 052525 → 152525
99	51.2	128											X			PCMMU CHECKOUT-052525 (OCTA L) 152525 → END

(1) Format size and frame length are mission-dependent (definition will be in SD78-SH-0044 Payload Carrier Mgmt FSSR) *

* The PCMMU subsets the downlist output by the GPC into this downlist window size for the 64 kbps downlink format.

NOTE: Telemetry format ID's 080-089 are assigned only to KSC to be used for independent data stream processing. No telemetry format ID's should be assigned that use the last two characters of this series of numbers.

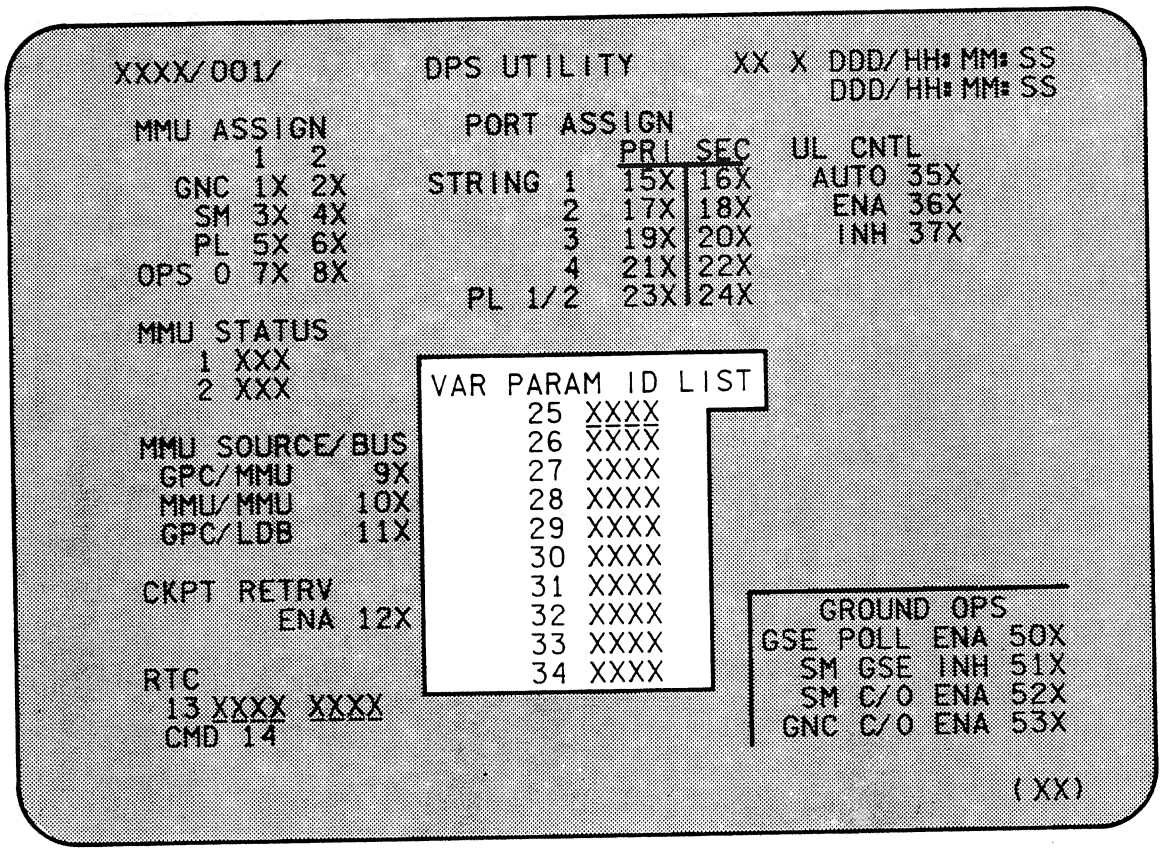


18821122B.ART; 2

DOWNLIST GPC. Item 44 (DOWNLIST GPC) allows manual selection of a GPC within the common set (CS) for downlisting. The GPC currently commanding PCMMU, TB 1 (the prime downlist buffer) is displayed, so the ID of a newly selected GPC will not appear until that GPC has taken control of the TB. ID's other than 1 to 5 will be rejected and ILLEGAL ENTRY displayed.

Item 49 (OPS 0 ENA) provides the capability to alternately enable and disable the downlist of a nonprime OPS 0 GPC. This item is initialized with the downlist disabled. An "*" is displayed next to item 49 when this item is enabled.

Figure 11.2.2-2.- Items 44 and 49 on GPC memory.



18821122C.ART; 2

Variable PARAM ID LIST. Variable parameters consist of a set of 10 memory locations (16-bit) that may be selected for incorporation into the downlist. These parameters are initialized to a nominal set of parameters that can be overwritten with new parameters via this SPEC or special UPLINK LOAD. Items 25 to 34 are used for this function. The new parameters are defined by entering the absolute main memory address in hexadecimal. Only memory core locations from 0000 to FFFF can be selected. Changes are effective when input and remain effective until changed.

Figure 11.2.2-3.- Variable parameters on DPS UTILITY display.

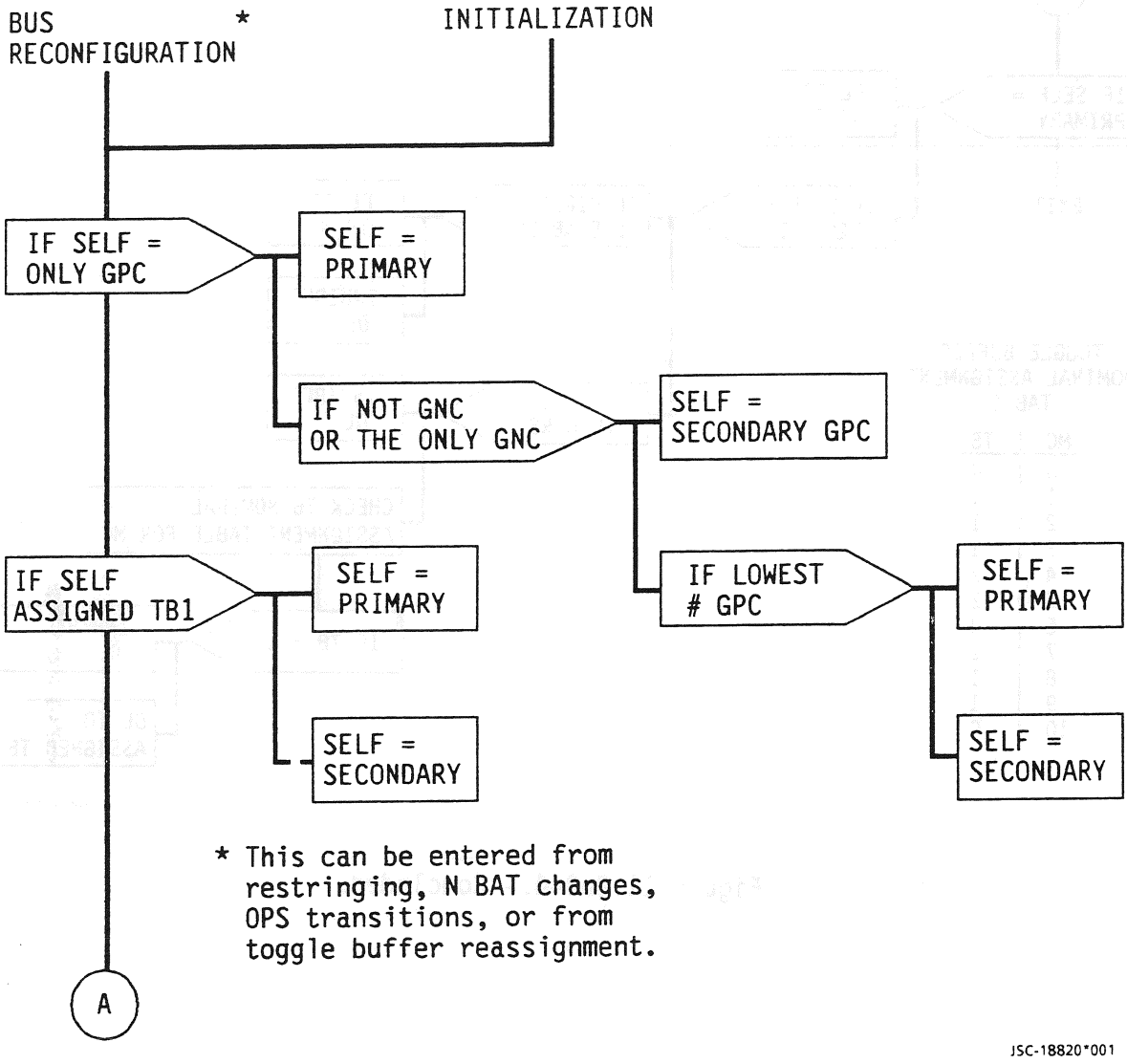
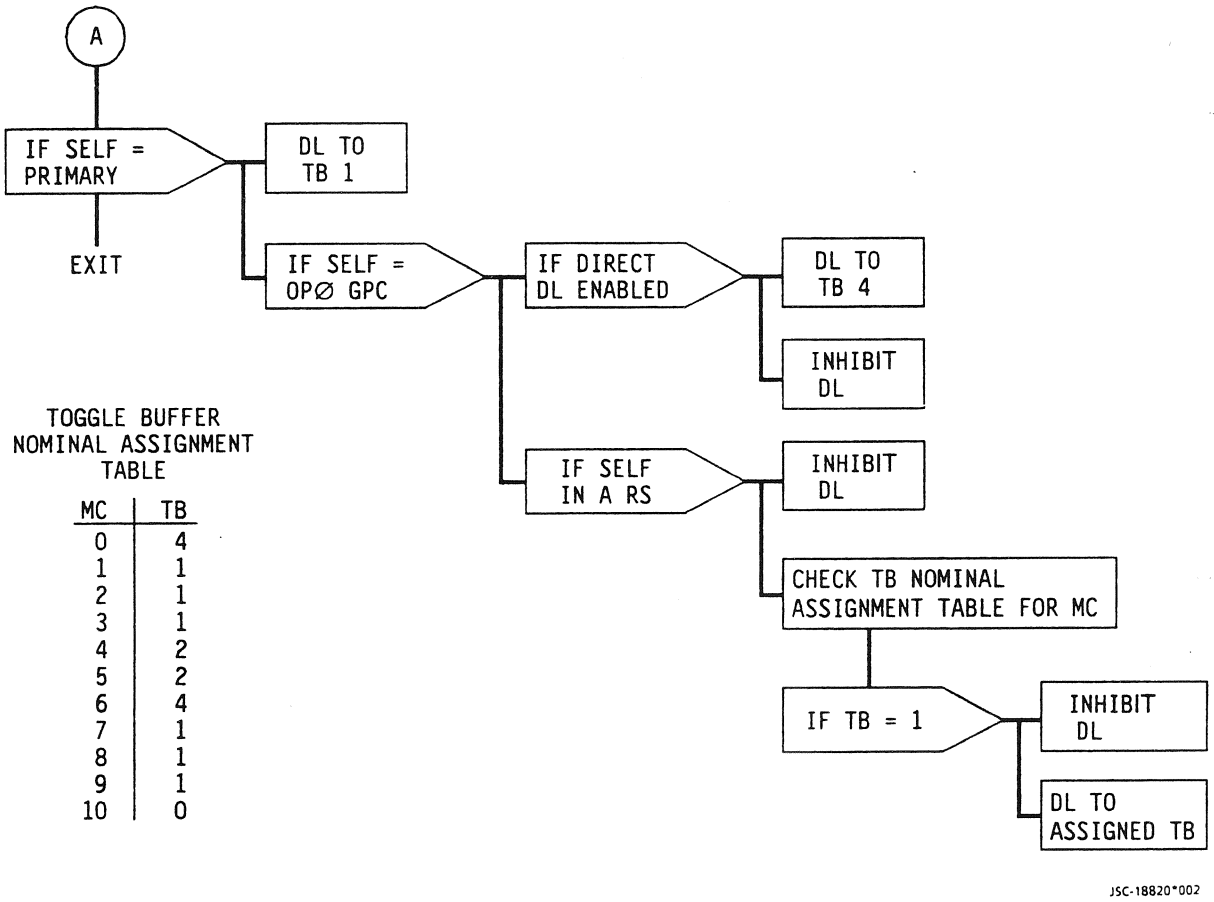


Figure 11.2.2-4.- Toggle buffer assignment.



TOGGLE BUFFER NOMINAL ASSIGNMENT TABLE

MC	TB
0	4
1	1
2	1
3	1
4	2
5	2
6	4
7	1
8	1
9	1
10	0

Figure 11.2.2-4.- Concluded.

11.2.3 GPC Software Dump[1]

11.2.3.1 Overview

The software dump allows any portion, or all, of a GPC main memory to be downlisted to the ground. The dump is controlled by software subroutines and software item entries made by the crew.

The software dump is used to dump the memory contents of a "good" GPC for one of two reasons. First, analysis of the dump can be used to confirm that the software loaded in the GPC is correct. Second, procedures require the crew to perform a software dump of a good GPC that was running at the time of another GPC failure. Analysis of this dump may provide information helpful in determining the nature of the failure of the bad GPC.

11.2.3.2 GPC Memory Spec

A software dump is initiated via crew inputs to the GPC memory spec (SPEC 0). See figure 11.2.3-1 for the GPC MEMORY display. Items 40 to 43 are used to request the dump, as follows:

A. Item 40 (START ID)

   Defines the absolute main memory address, in hexadecimal, of the start of the dump

B. Item 41 (NUMBER WDS)

   Defines the number of sequential 16-bit words of main memory to be dumped

C. Item 42 (WDS/FRAME)

   Defines the number of 16-bit words in the downlist frame to be used for the dump data

D. Item 43 (DUMP)

   Initiates execution of the dump via downlist, according to variables set in items 40 to 42

The software module ARF__DPS__CONFIG__ITEM processes the item inputs made to the GPC memory display. Figure 11.2.3-2 shows the detailed flow of this software module. This module checks the item number (ARF__NO) and performs the corresponding steps. The processing for the main memory dump items is as follows:

A. Item 40 (START ID)

The hexadecimal address input by the crew is saved in the downlist compool as the dump starting address. (CDWV__MAIN__MEM__PTR)

B. Item 41 (NUMBER WDS)

The decimal value input by the crew is saved in the downlist compool as the number of words to be dumped. (CDWV__MAIN__MEM__DMP__LEN)

C. Item 42 (WDS/FRAME)

The words/frame input by the crew is checked against the downlist word count (CZ1V__DOWNLIST__EOM). The downlist word count is equal to the frame length available for downlisting by the current memory configuration. If the words/frame input by the crew is greater than the downlist word count, an error message is annunciated. Otherwise, the words/ frame is saved in the downlist compool. (CDWV__MAIN__MEM__NO__WRDS__FRM)

D. Item 43 (DUMP)

If data has not been entered and accepted for items 40, 41, and 42, an error message is annunciated. Otherwise a check is made to ensure that the starting address (item 40) plus the dump length (item 41) is less than the maximum main memory address + 1, 211993. If the check is passed, then a flag in the downlist compool (CDWE__MAIN__MEM__DMP__REQ) is set to indicate a dump has been requested.

11.2.3.3 Dump Downlist

The downlist formatting software (DCDDOW) collects, formats, and transfers GPC downlist data. Processing that takes place for every frame of downlist may consist of three different types of data: main memory dump, post-IPL, and operational. Whenever a main memory dump is in progress, only the processing associated with it is performed. Figure 11.2.3-3 shows the detailed flow of this software module.

Main memory dump processing is invoked if the memory request flag (CDWE__MAIN__MEM__DMP__REQ) was set by SPEC 0 item entries. The request flag, as well as the starting address, dump length, and words/frame are available to the DCDDOW module from the downlist compool. All of these variables are initially set by items entries to SPEC 0 in ARF__DPS__CONFIG__ITEM.

Each time DCDDOW is called, it builds one frame of data for downlist. If a main memory dump is in progress, the frame will be filled with data words from main memory. These data words will be taken from the main memory starting address variable (CDWV__MAIN__MEM__DMP__PTR), up to the starting address plus the number of words per frame (CDWV__MAIN__MEM__NO__WRDS__FRM) minus the number of header words in this frame. Then the starting address

is incremented by the number of data words just put in the frame. The dump length (CDWV__MAIN__MEM__DMP__LEN) is decremented by the number of data words just put in the frame.

The starting address and dump length compool variables modified in this module are the same variables used for display purposes on SPEC 0. Therefore, as the starting address and dump length are modified after each frame, the values on SPEC 0 for items 40 and 41 will reflect the modification. The crew can watch item 40 count up and item 41 count down until the dump is completed.

11.2.3.4 Dump Formats

The Orbiter downlist can be transmitted at one of two rates. High data rate (HDR) is at 128 kilobits per seconds (kbps), and low data rate (LDR) is at 64 kbps. Each data rate is divided up among the various downlists - GNC, SM, PL9, BFS, OI, and PAYLOAD. The sum of the kbps for the individual downlists will equal the total kbps for the data rate.

The kbps is directly related to the words per frame based on the GPC output rate. The GPC puts out data at a rate of 25 GPC frames per second. The following equation specifies the relationship between kbps and words per frame:

$$\text{words/frame} = \frac{\text{kbps}}{25 \text{ GPC frames/sec} \times 16 \text{ bits/word} \times 1 \text{ kilobit/1000 bits}}$$

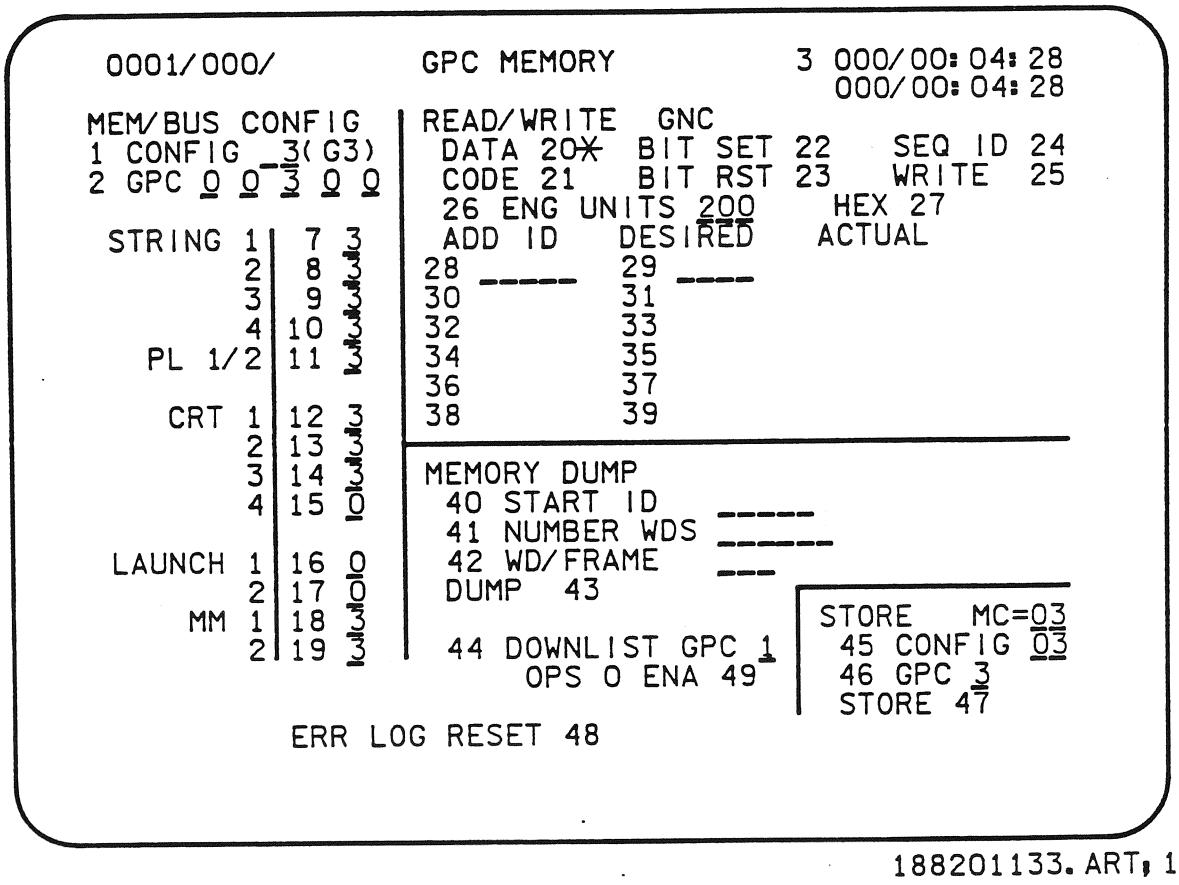
$$\text{words/frame} = \frac{\text{kbps}}{0.4}$$

This relationship is used to determine if the current telemetry format load (TFL) can support a dump at a specific words-per-frame rate. For example, if the crew is performing a software dump of the SM GPC and the current TFL has an SM bandwidth of 9.6 kbps, then the maximum number of words per frame is 9.6/.4, or 24. The crew must enter a number less than or equal to 24 for item 42, preferably equal to 24 so that the dump is downlisted as quickly as possible.

The format id for all software dumps is 93. The value is hard coded in the dump downlist software.

The actual contents of each frame of dump data is described fully in the HISAM Dump Software section, 11.1.3.

A detailed list of the words/frame for each orbiter telemetry format load (TFL) is contained in the Space Shuttle Telemetered and Recorded Data Format Requirements, JSC-10784, Rev K, January 1989.



188201133.ART; 1

Figure 11.2.3-1.- GPC MEMORY display.

11.2.3.5 References

1. Space Shuttle Program Orbiter Avionics Software Operational (OBS) Detailed Design Specification (DDS), vol. II, Systems Service, part III System Control (PASS Systems Software DDS).

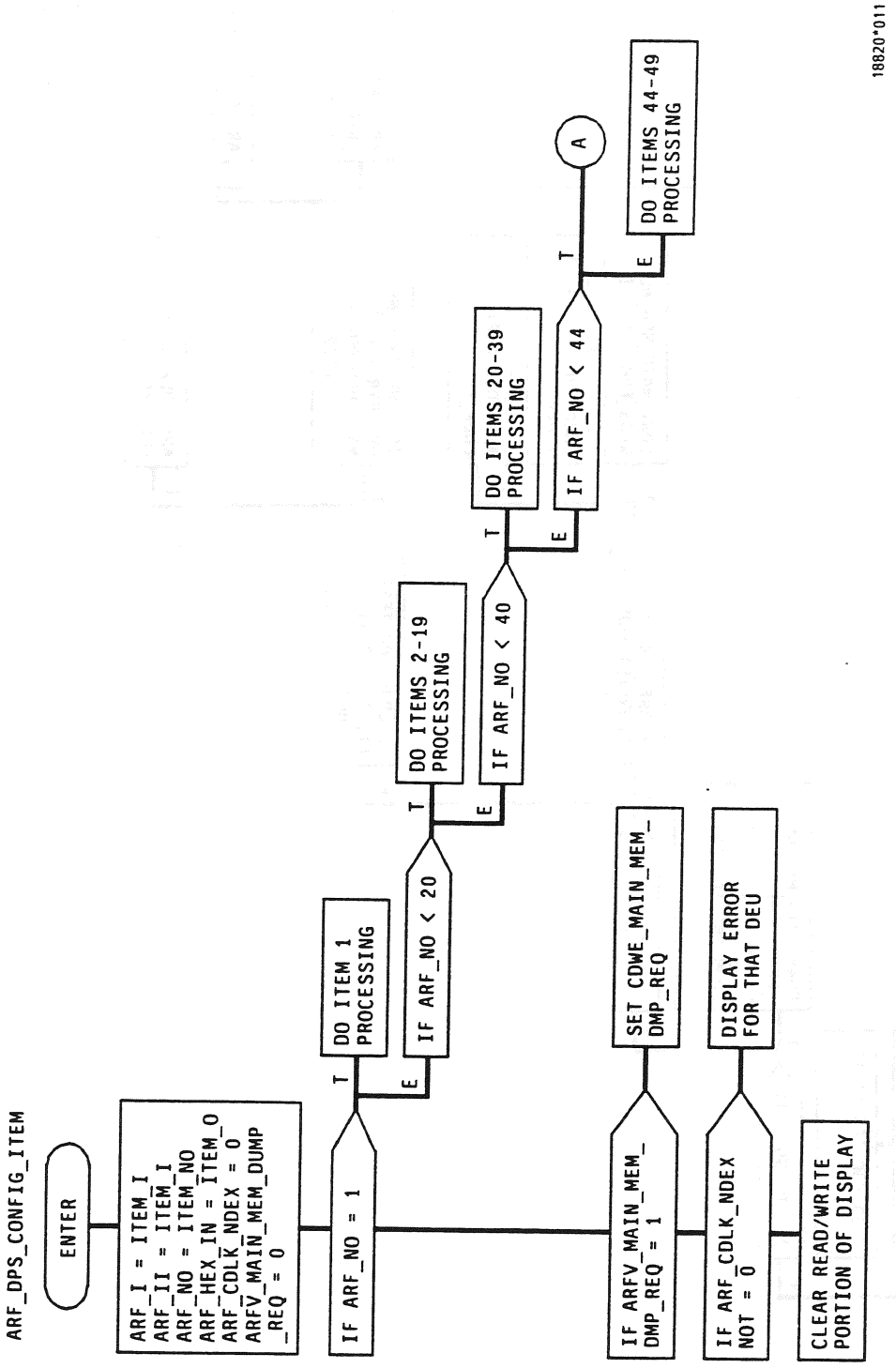


Figure 11.2.3-2.- PASS software dump item processor program flow.[1]

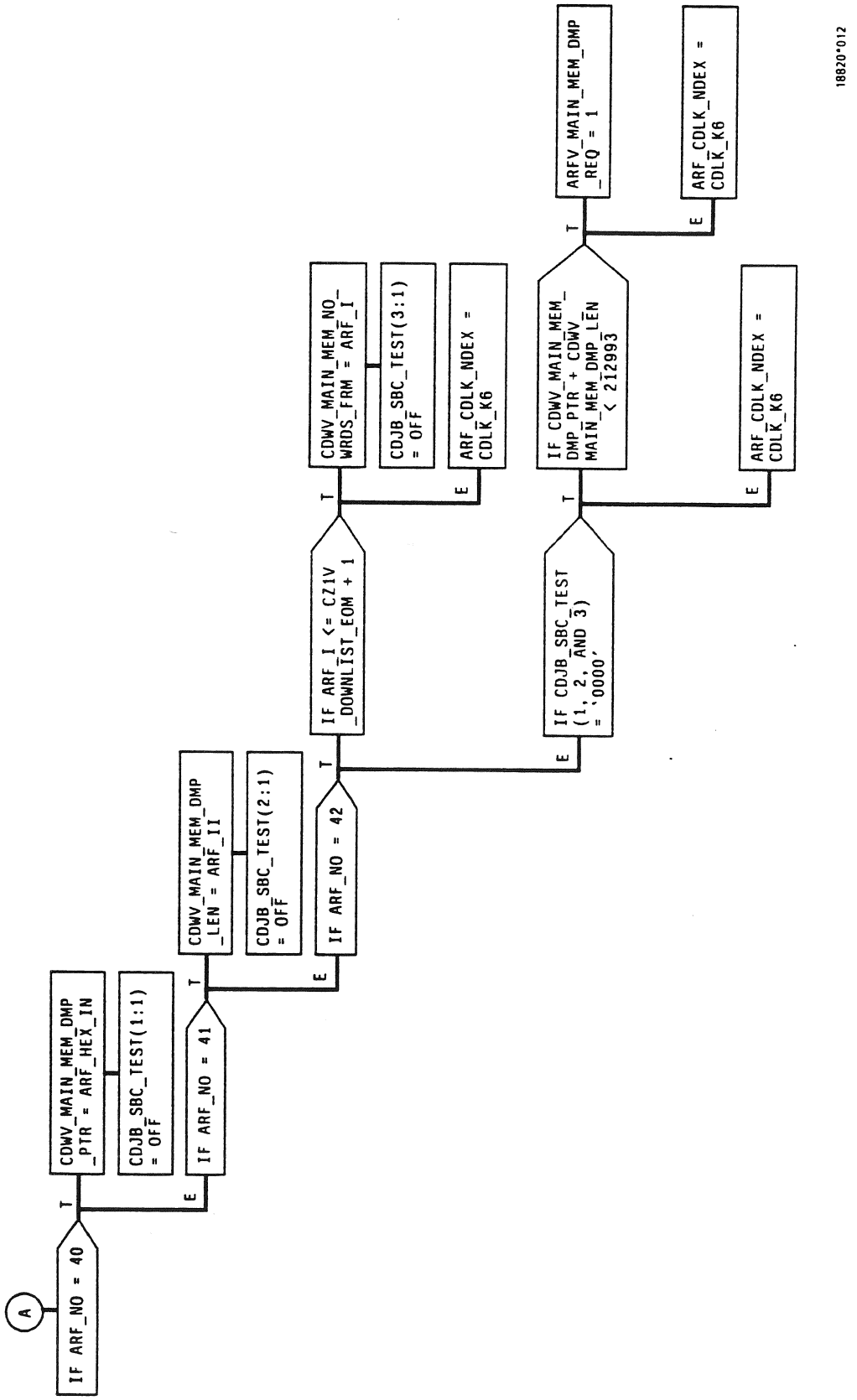


Figure 11.2.3-2.- Concluded.

TABLE 11.2.3-I.- ARF__DPS__CONFIG DEFINITIONS

ARF__I	- Item entry input value, if it is an integer
ARF__II	- Item entry input value, if it is a long integer
ARF__NO	- Item number
ARF__HEX__IN	- Item entry input value, if it is hexadecimal
ARF__CDLK__NDEX	- Error indication
ARFV__MAIN__MEM__DUMP__REQ	- Flag indicating main memory dump is requested
CDWE__MAIN__MEM__DMP__REQ	- Compool variable used to indicate main memory dump is requested
CDWV__MAIN__MEM__DMP__PTR	- Compool variable containing the starting address of the dump
CDWV__MAIN__MEM__DMP__LEN	- Compool variable containing the length of the dump
CDJB__SBC__TEST	- Blanking words for the items on the CRT display
CZ1V__DOWNLIST__EOM	- Downlist word count for EOM building
CDWV__MAIN__MEM__NO__WRDS__FRM	- Compool variable containing the number of words per frame for the dump

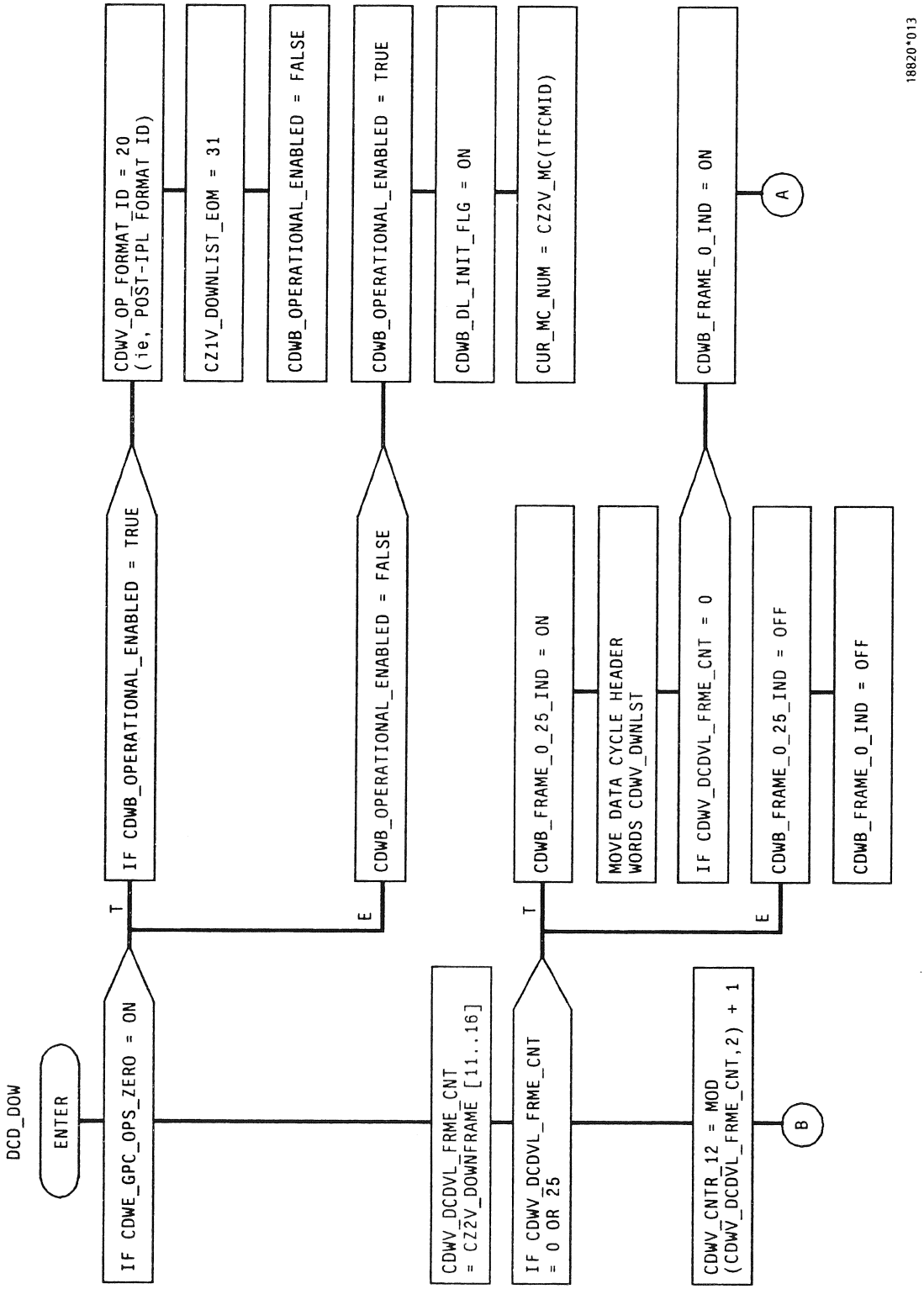


Figure 11.2.3-3.- PASS software dump downlist program flow.

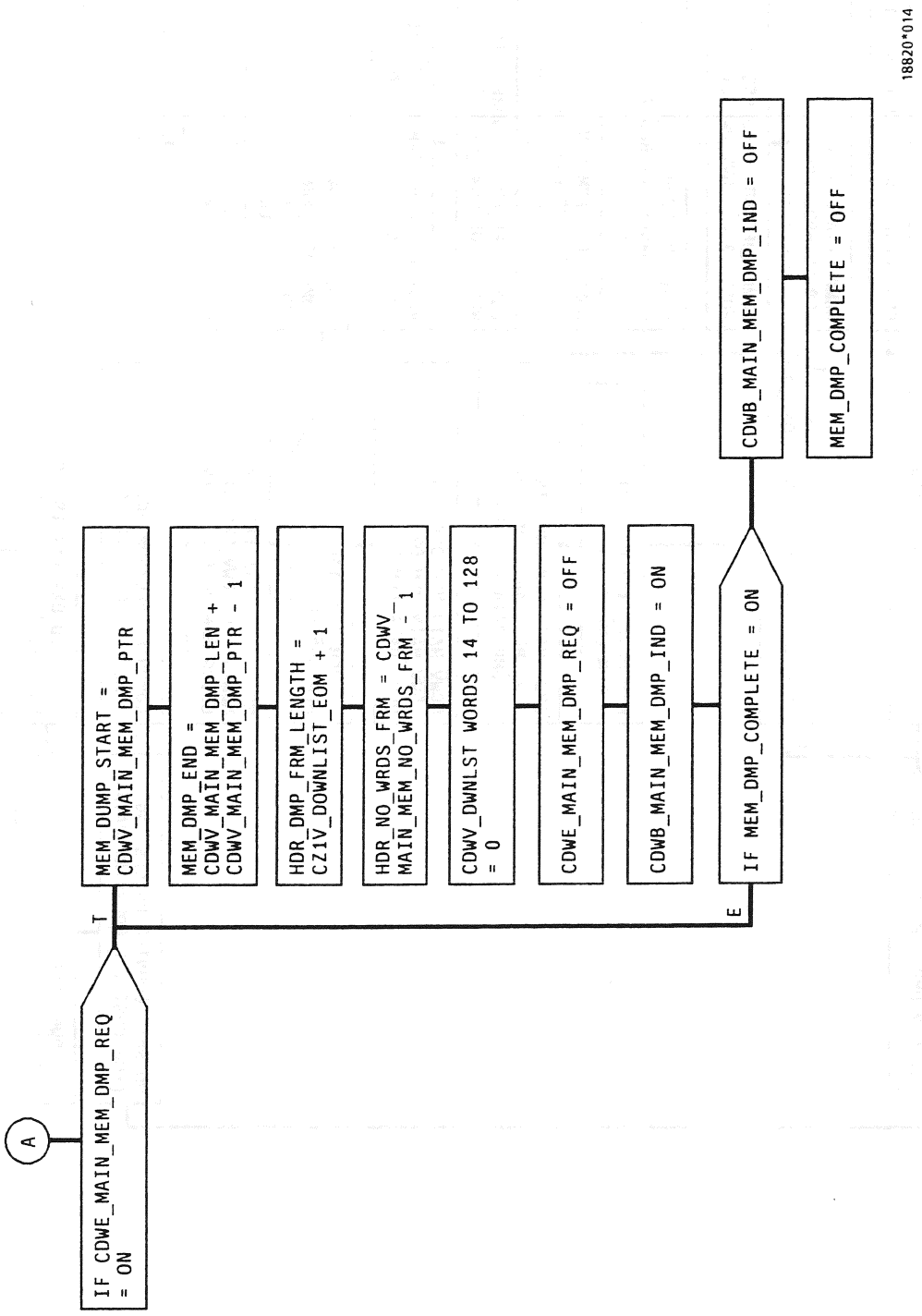


Figure 11.2.3-3.- Continued.

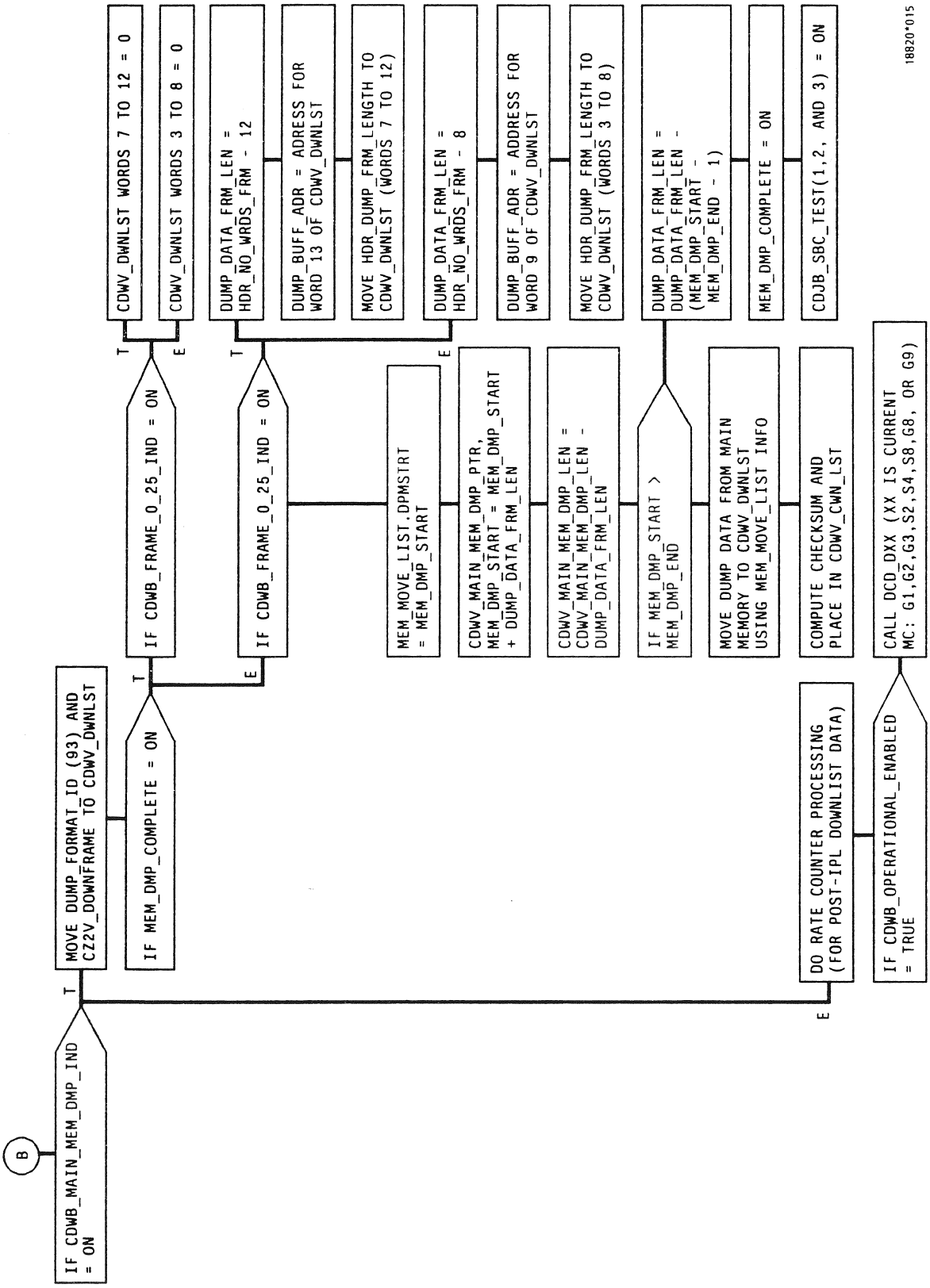


Figure 11.2.3-3.- Concluded.

TABLE 11.2.3-II.- DCD__DOW DEFINITIONS

CDJ__RW__DATA definitions:

CDJB__SBC__TEST	- Blanking words for items on CRT display

CDW__DOWNLIST__COMPOOL definitions:

CDWB__DL__INIT__FLG	- Initialization flag for operational downlist formatter
CDWB__FRAME__0__IND	- Indicates frame 0 is being processed
CDWB__FRAME__0__25__IND	- Indicates frame 0 or 25 is being processed
CDWB__MAIN__MEM__DMP__IND	- Indicates main memory dump in progress
CDWB__OPERATIONAL__ENABLED	- Indicates operational processing is active
CDWE__GPC__OPS__ZERO	- Indicates no active overlay
CDWE__MAIN__MEM__DMP__REQ	- Indicates main memory dump is requested
CDWV__CNTR__12	- Indicates which set of 12.5 s/s parameters is being downlisted
CDWV__DCDVL__FRAME__CNT	- Frame number (shadow of CZ2V__DOWNFRAME)
CDWV__DWNLST	- Downlist output buffer
CDWV__MAIN__MEM__DMP__LEN	- Current number of words remaining to be dumped (displayed on SPEC 0 by item 41)
CDWV__MAIN__MEM__DMP__PTR	- Starting address for the current frame of data to be downlisted (displayed on SPEC 0 by item 40)
CDWV__MAIN__MEM__NO__WRDS__FRM	- Number of words per frame
CDWV__OP__FORMAT__ID	- Current format id

TABLE 11.2.3-II.- Continued

CZ1V__COMMON compool definitions:

CZ1V__DOWNLIST__EOM	- EOM address of downlist frame
CZ2V__DOWNFRAME	- Contains the downlist frame number in bits 11 to 16
CZ2V__MC	- Current memory configuration

Local variable definitions:

CUR__MC__NUM	- Current memory configuration number
DUMP__BUFF__ADR	- Address of current location in downlist buffer (equivalent to MEM__MOVE__LIST.DPMDATI)
DUMP__DATA__FRM__LEN	- Number of data words in a frame, not including header words (equivalent to MEM__MOVE__LIST.DPMLEN)
DUMP__FORMAT__ID	- Equals 93
HDR__DMP__FRM__LENGTH	- Frame length, to be included in header
HDR__NO__WRDS__FRM	- Total words in a frame, including header words
MEM__DMP__END	- Ending address for memory dump
MEM__DUMP__START	- Starting address for memory dump
MEM__DMP__COMPLETE	- Indicates memory dump has competed
MEM__MOVE__LIST.DEMPSTRT	- Dump starting address

11.3 SYSTEM CONTROL[1]

System Control is one of three major activities that takes place under Systems Services as illustrated in figure 11.3-1.

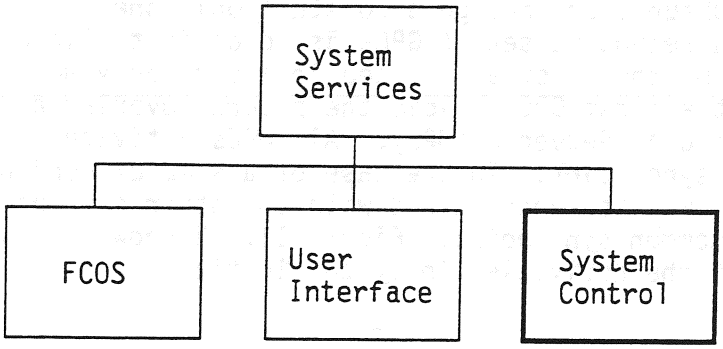


Figure 11.3-1.- Overview of Systems Services.

The functions of System Control include System Initialization which is initialization of the system following the loading of a GPC via IPL, System Reconfiguration which handles things like restringing, and System Specialist Functions which are system oriented rather than directly related to the major function of GNC, SM or PL. Figure 11.3-2 presents the activities that are managed by System Control which includes System Initialization, System Reconfiguration, and System Specialist Functions.

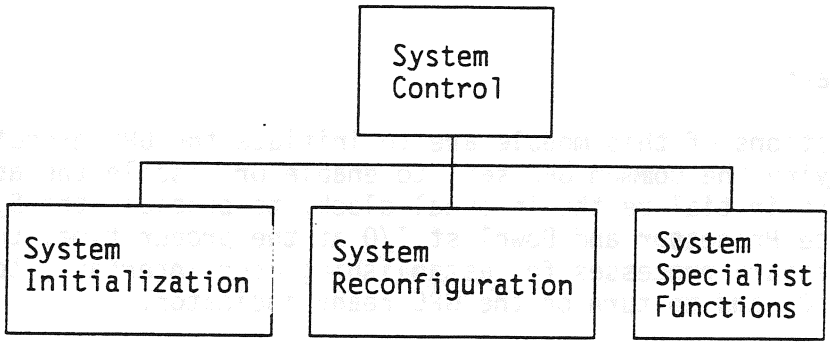


Figure 11.3-2.- System Control.

A. System initialization

The System Initialization software initiates the execution of User Interface and System Control software and places the GPC into a state from which normal operation may begin. This is accomplished through a series of subfunctions designed to start only one GPC at a time. In the case where a redundant set of GPCs is to be initialized, all subsequent GPCs following the first are placed in a secondary mode and await a command by the first GPC to load the program overlay and initiate the first Operational Sequence (OPS). All GPCs activated are synchronized at a common sync point. In the case of a simplex configuration initialization, the GPC is synchronized with any other currently executing GPC set at the common sync point. Figure 11.3-3 shows the detailed breakdown of the activities in System Initialization.

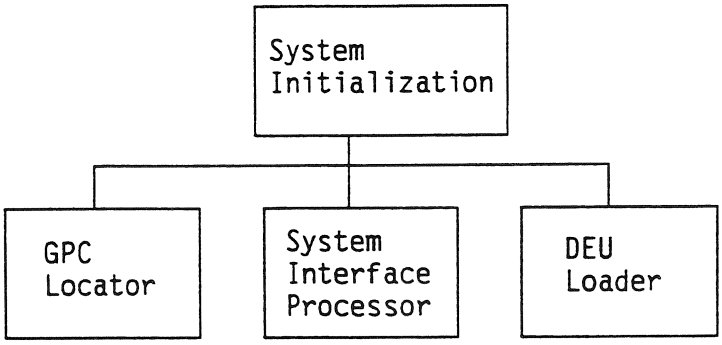


Figure 11.3-3.- System Initialization.

1. GPC locator

The functions of this module are to initiate the GPC execution by identifying the common GPC set, to enable or disable the appropriate buses, to initialize the internal clock, to schedule the System Interface Processor and Downlist I/O at the proper time, to schedule the necessary processes for establishing user interface, to initiate GPC OPS 0, and to turn on the GPC ready indicator.

The initial sync is issued to test for the existence of a common set. A series of data items are initialized and various events are reset, after which the ICC and PCMMU buses are enabled. If there is not a common set, then the Primary GPC Initialization is performed. Otherwise, Secondary GPC Initialization is performed.

Primary GPC Initialization starts with more data items being initialized. The Flight Critical data buses, DK 1-3, Payload data buses, and the Launch Data buses are enabled. The GPC's internal clock is initialized, the System Interface Processor is scheduled and the Correct Moding and BFS SVC (sets bus marks on PL buses if BFS is in RUN, reads BFC CRT display discretes, sets DK bus mask as

appropriate, and reads the engage and/or I/O Term B discretes to determine FC) is issued. Finally, the toggle buffer is assigned and downlist I/O is initiated. Annunciation Processing will be cleared and LDB processing will be initialized according to its previous state.

Secondary GPC Initialization starts with the Flight Critical, DK, Payload, and Launch Data Buses disabled. The System Interface Processor is scheduled temporarily to establish communication with the other GPCs in the common set so that data items in the compools can be initialized. After the data has been initialized and the System Interface Processor has cancelled itself, the GPC's internal clock is initialized and the System Interface Processor is scheduled permanently. After a 2-second delay, the toggle buffers are assigned. The downlist I/O is initiated and the Correct Moding and BFS SVC is issued. Finally, if the current common set commands no DEU (including DEU 4), then DEUs 1 to 3 are assigned to the GPC that was just added to the common set.

The output talkback is changed to gray, assuming that I/O Term B is not set. A pseudo keyboard command is generated to start OPS 0. The processes necessary for establishing user interface are scheduled and the GPC ready indicator is turned on.

2. System interface processor

This module provides control of system wide processes which are required to operate at the same time and in a coordinated manner. It allows multiple GPCs to communicate in an efficient manner and to participate in GPC and MTU redundancy management processing.

The System Interface Processor calls the Downlist Processor to fill the downlist buffer. The remainder of the processing takes place in four phases numbered 0 through 3.

a. Phase 0

If the message routing from the previous cycles Phase 3 was not performed, then message routing is performed in this Phase 0. The message routing process is described in Phase 3.

If message routing is not required and this is the first minor cycle of a new major cycle, then annunciation collection is performed. This involves switching pointers to annunciation buffers, processing the last major cycle's annunciation bits, and building an ICC message to be sent during the next minor cycle. NSP timer initiated I/O is also performed.

b. Phase 1

The NSP data read during the previous Phase 0 is moved into the ICC buffer from the NSP input buffer for the current NSP. If

the bus mask is set, or the BCE has been bypassed for the current NSP, an attempt is made to restore the other NSP element. If the last ICC buffer load has been routed, then the next ICC buffer load is collected. If an annunciation message is ready, then it is loaded first. Next, the messages from the ICC Collector Buffer are loaded if they all fit. Finally, the ICC Status Flags are scanned. For the first three of these that are found set, the corresponding ICC message is constructed and loaded until the buffer is full. The end of the data indicator is saved and the remaining messages will be handled as described in Phase 0.

c. Phase 2

No additional processing is done during this phase during which the ICC I/O is taking place.

d. Phase 3

The ICC buffers of all members of the common set are processed sequentially. If NSP data is present, then it is routed to the NSP compool unless it was found in a previous buffer or the processing of the previous data is not yet complete. Then messages in the ICC buffer are routed. The header for each message is decoded and the appropriate action taken (procedure called, data moved to compool, or event set). This process continues until all messages in the buffer have been processed. For some messages, only the first occurrence is to be processed. The decoded message header identifies these filtered messages and the filtering logic screens multiple occurrences of the same message in subsequent ICC buffers. Finally, after all message routing has taken place, the MTU Update SVC is issued if this is SIP cycle 19.

Errors in ICC messages will result in a GPC error log and entry. The remainder of the ICC buffer is not routed.

3. DEU Loader

DEU Loader provides the capability to load the DEU when the GPC is in Post IPL OPS 0 or PL OPS 9.

The DEU Loader determines which DEU is to be loaded from the DEU Load Table. A BITE status request is issued to determine if the IPL PROM was successful. A DEU load is retrieved from the mass memory unit and buffered in DEU Memory Fill Buffer. Blocks are transferred to the DEU via "memory fill" messages. A delay of 18 ms between each block transferred occurs so that the DEU may transfer the block to its proper location. There is a 300 ms delay after the last block is transferred, then a Mode Status and BITE Status Request is issued to determine if the IPL was successful.

If the IPL was successful, the PASS Critical Formats are retrieved from mass memory and buffered into the DEU Memory Fill Buffer. Critical formats are transferred to the DEU via "format data fills." A delay of 18 ms occurs between format data fills to allow the DEU to transfer the data to its proper location. Following the last format data fill and a delay of 70 ms, a Mode Status Request is issued to determine if the critical format load was successful. If there is no checksum error on the Mode Status Request, the critical format load was successful.

If an error is detected at any point in the DEU loader's sequence, the process is terminated and an error message is issued.

B. System Reconfiguration

The System Reconfiguration software coordinates changes in the makeup of the DPS. Included are changes in the GPC set, main memories, and IOP channels, as well as supporting the display of configuration status and user input. The operational sequence for system idling as a major function or as an entire GPC is included in this set of system services. It also provides the capability to monitor the GPC mode switch, the BFC engage discretes, and other selected discretes. Figure 11.3-4 shows the elements that make up system reconfiguration.

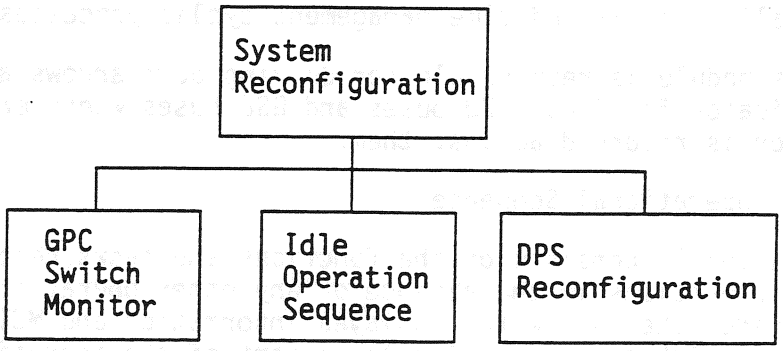


Figure 11.3-4.- System Reconfiguration.

1. GPC Switch Monitor

The functions of this module are to monitor the Mode switches, the BFS engage pushbutton, the Output switches, the BFC CRT Select switch, and to compute values for the GPC/Bus Status display.

The discretes to be monitored are read and redundancy managed. The BFS Run discretes are monitored for the Run status. The Payload Data Bus masks are set when the BFS is in Run. If a PASS GPC Mode switch has changed from Run to Standby and then back to Run, then the GPC is forced to OPS 0. The status of the Output switch and the BFS engage switch is compared to the previous status. If a PASS GPC

I/O Terminate B discrete is set, the Output talkback indicator will go barberpole and the flight critical bus masks will be set. The DEU mask is set for the CRT selected by the BFC CRT Select discretes. Any DEUs that are being reclaimed from the BFS will have a major function change forced on them to match the currently selected major function.

The GPC votes against each GPC are tabulated and any computer that has two or more votes against it will have its failed indicator set so that a down arrow will be displayed as it status. The characters representing a major function and OPS are computed for each GPC and stored for display purposes. The error counter for each DEU is interrogated, and those with a count greater than one will be flagged with a down arrow. The GSE buses are flagged in a similar manner.

There is a special check made on all common set (CS) GPC's, which can force a GPC's duty cycle to zero. A GPC's duty cycle is forced to zero if: (1) it is no longer in the CS, (2) it is not an OPS 0 GPC, and (3) it has dropped its sync lines [000]. This mechanization allows us to detect a GPC failing to the halt state (FTH GPC), or a GPC which has failed-to-sync (FTS) or failed-to-quit (FTQ) being taken to STBY/HALT.

Finally the NSP and time management cyclic processes are called.

This module is responsible for driving down arrows as a status indicator for GPCs, DEU buses and GSE buses whenever more than one error is recorded against them.

2. Idle Operational Sequence

This module consists of the functions and tasks needed to satisfy DPS operations in the absence of any other operational sequence. It provides the user with displayed information and MCDS keyboard entry items which are not available as part of the operational sequence associated with the major functions (except as SPEC 0 or MODE 0). This module is illustrated in figure 11.3-5.

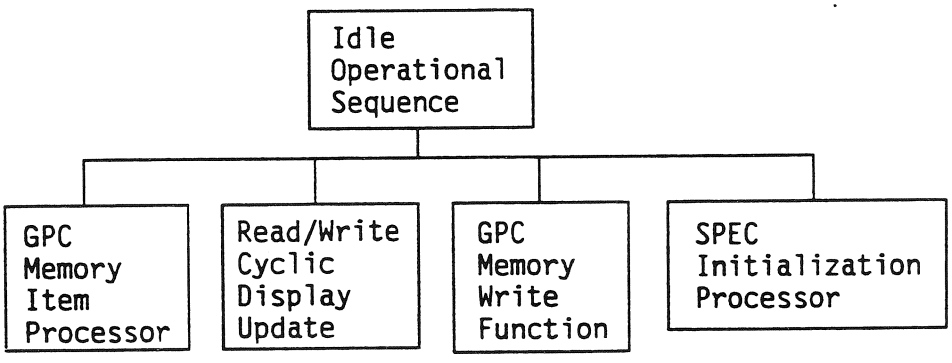


Figure 11.3-5.- Idle Operation Sequence.

Upon display initialization, the SPEC Initialization Processor is called. After this initialization, the Read/Write Cyclic Update program is scheduled. The display programmer macro is processed and the GPC Memory Item processor is called.

a. GPC Memory Item Processor

The GPC Memory Item Processor services the item inputs made on the display GPC Memory (OPS 0, Spec 0). This module handles all of the item entries made to this page and determines whether they are valid or not.

b. Read/Write Cyclic Display Update

The Read/Write Cyclic Display Update module contains the logic required to provide for the cyclic updating of the "ACTUAL" data fields on the Memory Read/Write portion of the GPC MEMORY display. The cyclic rate used is once every 2 seconds.

c. GPC Memory Write Function

The GPC Memory Write Function contains all the software required to update GPC memory based on user input.

This module is scheduled whenever the user initiates a GPC Memory Write request while the GPC Memory display is active.

d. SPEC Initialization Processor

The SPEC Initialization Processor is called to initialize data for the GPC Memory and DPS Utility SPECs.

3. DPS Reconfiguration

The DPS Reconfiguration software is responsible for the reconfiguration of GPC sets, main memory reconfiguration, changes to the GPC Reconfiguration Table, and changes to bus configuration and its associated table. Figure 11.3-6 contains the modules that make up the DPS Reconfiguration.

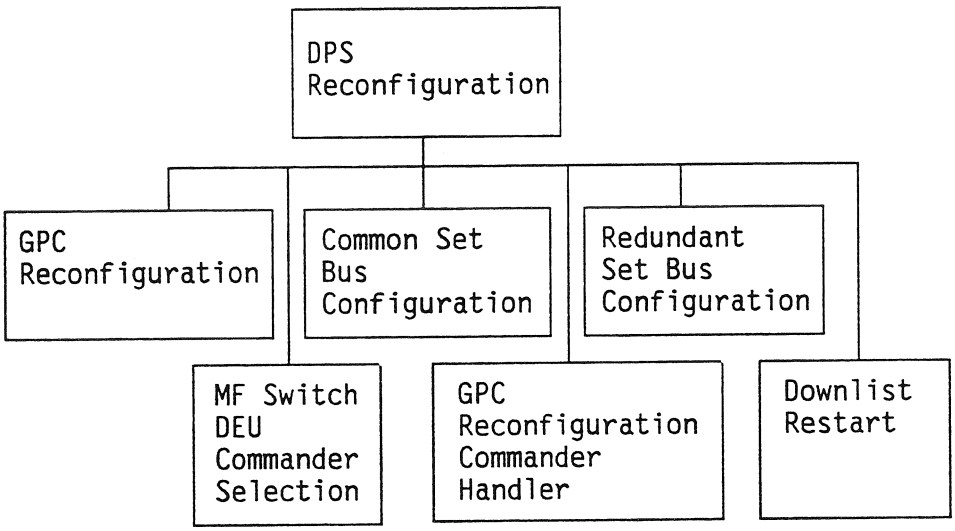


Figure 11.3-6.- DPS Reconfiguration.

a. GPC Reconfiguration

The GPC Reconfiguration module provides the capability to service a request to overlay main memory with a new memory configuration. This function may be done in conjunction with a redundant set formation or standalone with or without a new OPS initiation. A new OPS may be initiated or the GPC may be placed into a ready state for further crew action.

This function provides the only process where the Memory Configuration of a GPC may be modified aside from that provided by the System Software Loader. This function may modify the Memory Configuration of a GPC in the run mode either as a simplex GPC or with other GPCs to form a redundant set. An overlay is required for some OPS transitions, but not for all. The basis for making the decision to overlay or not is the GPC Reconfiguration Table and the GPC set as a function of the Memory Configuration. The source of a Memory Configuration which is controllable by the crew through the DPS Utility display may be another GPC or from the Mass Memory Unit.

Three situations require main memory overlay:

(1) The OPS requested is not contained in the current MC.

(2) An OPS is requested which requires formation of a new redundant set which differs from the original set.

(3) Crew-set parameters force an overlay upon OPS request regardless of current memory contents (this is done via the MMU/MMU setting on the DPS Utility display).

There are four valid entry codes for GPC Reconfiguration with values as follows:

0 - Request for reconfiguration
2 - Freeze Dry request
5 - Request to pre-position the mass memory for the upcoming OPS request
10 - Request for reconfiguration in secondary GPCs

Entry type 2 is used by the GPC Memory Item Processor to service a Freeze Dry request. A Freeze Dry entry allows for the loading of a MC into a GPC without initiating execution of the associated application OPS. GPC reconfiguration will initiate the loading of the major function base, if not already resident in main memory, and the OPS overlay for the target MC. Upon successful completion of the Freeze Dry Load, the GPC will be forced to OPS 0.

Entry type 5 is used by the UI Control Supervisor to pre-position both Mass Memory Units prior to the actual OPS transition in order to minimize the OPS downtime during OPS transitions. If the crew has selected GPC/LDB for overlay source (via DPS Utility display), then no pre-position processing is performed. Otherwise, both MMUs will be pre-positioned to the subfile preceding the required overlay data. If neither MMU can be pre-positioned successfully, then the OPS transition request is rejected. However, if at least one MM was successfully pre-positioned, then the OPS transition processing continues.

Entry type 0 and 10 are used for primary and secondary GPC overlay processing respectively. The initial processing is done for each, as these two entries differ (this is done prior to the new RS formation). However, just prior to the formation of the reconfiguration RS, the processing for these two entries converges. The processing is unique to each of these two entries and will now be described.

The type 0 entry is used by the Sequence Request Processor for the GPC (or GPCs in a RS) receiving the OPS transition request (i.e., the primary GPCs). Upon receiving the control, GPC reconfiguration computes the data required for overlay processing:

(1) RS mask including current RS, target RS, and a mask of secondary GPCs.

(2) Overlay source/destination masks

(3) A reconfiguration RS mask, which includes both current target GPCs.

Using this data, a type 1 ICC message is built and set to those GPC's that are required in the target RS but are not in the current RS (these are the secondary GPCs). The data in this ICC message will enable the secondary GPCs to join the reconfiguration RS and participate in the OPS transition.

The type 10 entry is used by the GPC reconfiguration Message Handler to prepare the secondary GPCs for RS formation. The data from the type 1 ICC message noted above, is used for the preparation of the secondary GPCs so that their overlay related data will be consistent with that of the primary GPCs.

Following the processing unique to entry types 0 and 10, the primary and secondary GPCs continue with identical processing (note that the RS that exists at this point is the RS which detected the OPS transition request). The processing begins with the cancellation of the following programs:

(1) GPC Switch Monitor

(2) Cyclic Display Processor

(3) Read/Write Cyclic Display Update

(4) MCDS input processor

(5) LDB I/O Processor

(6) NSP Message Processor

Once these programs are cancelled, the data required for these programs to run redundantly within the reconfiguration RS is initialized.

At this point, everything is set up so that the reconfiguration RS can be formed. In order to do this, the primary GPCs issue a type 2 ICC message, which brings the primary and secondary GPCs into sync so that the reconfiguration RS can be formed. Once the new RS is formed, the programs which were previously cancelled are re-scheduled and processing proceeds to overlays.

Before overlay processing begins, any GPCs which are in the reconfiguration RS and are not target GPCs are removed from the reconfiguration RS. All target GPCs then do overlay processing, starting with GPC-to-GPC overlays and moving on to Mass Memory overlays. Any GPCs which experience errors on the GPC-to-GPC overlays are dropped from the reconfiguration RS and are taken out of the target set. Only those GPC's in the target set will attempt the actual overlay. After the Mass Memory overlays are

completed, any GPC's which had overlay errors are dropped from the reconfiguration RS. The reconfiguration RS at this point, is made up of all target GPC's which did not experience overlay failure.

b. Common Set Bus Configuration Change

Common Set Bus Configuration Changes provide a method of transferring bus commands from GPC to GPC. It is invoked as a result of crew inputs or from processing associated with OPS transitions or removal of a GPC from the redundant set.

Each value in the bus configuration change request ICC message, is limit-checked to verify that the value is reasonable. If any ICC message value fails the limit check, the ICC Limit Status Flag is turned on and no further processing is done for the request.

Bus reconfiguration is accomplished by issuing the RECONFIG SVC. Each GPC in the Common Set, participates in the bus assignment by issuing the SVC with the appropriate request type. Processing in the Common Set Bus Configuration Change is divided into five major areas, one for each bus configuration request type. Each of these areas are discussed below.

(1) Major Function Switch Change - The input DEU is assigned to the specified New Bus Commander, which is selected by the MF switch DEU Commander Selection.

A zero New Bus Commander indicates that the DEU's major function is not active in the common set, so the DEU should remain with its current commander.

The Current MF for the DEU is updated to reflect the DEU's new major function setting.

GPCs that detected the major function switch change complete the major function switch change "house-keeping" processing that was initiated by the DEU polling function.

(2) GPC/CRT Key Request - If the specified DEU is being reassigned from isolation (i.e., not commanded by a GPC in the Common Set), to a New Bus Commander who is a member of the Common Set, the number of DEUs already commanded by GPCs in the Common Set is determined by counting the number of DEUs not indicated in the isolated DEU mask. An error message is issued and no further processing is done if the DEU reassignment would cause more than three DEUs to be commanded by the Common Set.

Assigning a DEU to a GPC via the GPC/CRT key, causes the GPC/CRT Commander field to be updated in the new commander's RS.

(3) FC/PL Mode Change or GPC Prime Reassignment - An Action Code of 5 indicates that Prime reassignment is to be performed. This is accomplished by updating the GPC prime ID to the ID Code Value and invoking Bus Reconfiguration to do toggle buffer assignment processing.

An Action Code value other than 5 indicates that a FC/PL mode change is to be done. The ID code, which specifies the string number (string 5 represents the Payload buses), and the Action Code, which specifies the mode, are passed to the Bus Reconfiguration where the mode change processing is performed.

(4) Reassign GSE Polling - Bus Reconfiguration is invoked to reassign the specified GSE polling bus to the new polling GPC. If GSE polling is enabled for the DPS and self is a member of the new polling GPC Redundant Set, then the current state of GSE polling for self is examined. If GSE polling is disabled for self, polling is enabled for self and the LDB I/O Processor is Scheduled. Otherwise, if GSE polling is disabled for the DPS, or self is not a member of the new polling GPC Redundant Set, then GSE polling is disabled for self.

(5) LDB Bus Reassignment - Bus Reconfiguration is invoked to reassign the specified LDB bus to the specified New Bus Commander. However, if the specified New Bus Commander is 6, then the New Bus Commander is determined from the Nominal Bus Assignment Table.

An optional second LDB bus, if provided, is processed in the same manner.

c. Redundant Set Bus Configuration Change

Redundant Set Bus Configuration Change provides bus configuration change processing on a redundant Set or Simplex basis. It is invoked as a result of crew inputs for Mode Recall, OPS Transition, or taking a Redundant Set to OPS 0.

Bus reconfiguration is accomplished by issuing the RECONFIG SVC. Each GPC in the participating set, issues the SVC with the appropriate request type. Processing in Redundant Set Bus Configuration Change is divided into four major areas. Each of these areas is discussed below.

(1) Mode Recall (with no set change) - If any entry in the Nominal Bus Assignment Table for the currently active

memory configuration is different from the corresponding value obtained from the Nominal Bus Assignment Table at the previous OPS Transition (or subsequent Mode Recall), then the flight critical strings, the Payload buses, and the LDB buses will be reassigned according to the updated Nominal Bus Assignment Table. DEU processing will be identical with that performed for OPS Transition Final Processing (major area 4, described below).

(2) OPS Transition (or Mode Recall with set change) Initial Processing - If the set is not changing and the values in the Nominal Bus Assignment Table for the currently active memory configuration are identical with the values obtained from the Nominal Bus Assignment Table at the previous OPS Transition (or subsequent Mode Recall), then the only processing done, is to (1) simulate major function switch processing for all DEU's commanded by the participating set in order to satisfy the User Interface Control Supervisor, and (2) Schedule the LDB I/O Processor, and (3) clear the GPC/CRT Commander field for all DEUs.

Otherwise, either the set or the Nominal Bus Assignment Table values being used have changed, so the following processing is performed. If the participating set is a Redundant Set, all DEUs commanded by GPCs in the Redundant Set are temporarily reassigned to their current commanders to ensure that receivers are on and bus masks are off for the DEUs in all GPCs in the (possibly new) Redundant Set. Also, if the redundant set has changed, the lowest GPC in the new Redundant Set is chosen as Prime. If the OPS Transition is for a simplex GPC, the GPC is assigned a toggle buffer as specified by the Toggle Buffer Nominal Assignment Table.

Regardless of whether or not the participating set is also a Redundant Set, the flight critical strings, the Payload buses, and the LDB buses are reassigned according to the Nominal Bus Assignment Table. MTU and NSP BCE elements are restored.

Major function switch processing is simulated for all DEU's commanded by the participating set in order to satisfy the User Interface Control Supervisor.

Processing for GSE Polling is performed at every OPS transition. If GSE Polling is enabled for the DPS, then GSE Polling is enabled or disabled for GPC Self depending on whether or not the participating set commands at least one LDB bus. If the participating set commands at least one LDB bus GSE Polling will be enabled for GPC Self. Otherwise, it will be disabled. The LDB I/O Processor is always Scheduled, even if GSE Polling is disabled (the LDB

I/O Processor will do cleanup processing and then terminate itself).

(3) Redundant Set to OPS 0 - All Flight critical strings commanded by GPCs in the old Redundant Set are reassigned to the lowest GPC in the old Redundant Set. Receivers and bus masks are updated for the DEU, Payload, and LDB buses by assigning each bus to self (if self is the bus's commander) or GPC 6. If GPC Self is not the commander of the current GSE Bus, then the GSE Poll State is turned off. Major function switch processing is forced for all DEUs commanded by GPCs in the old Redundant Set.

(4) OPS Transition (or Mode Recall with set change) Final Processing - Cleanup is done for items 54 and 55 on SPEC 1. DEU reassignment processing is done for each DEU which is in the same MF as the participating RS.

d. MF Switch DEU Commander Selection

MF switch DEU Commander Selection logic selects the new commander for a DEU for which a major function (MF) switch change has been detected. The DEU's new commander is selected by the GPC (or Redundant Set) that detects the MF switch included in the MF switch ICC message so there will be no disagreement among Common Set GPCs about which GPC should command the DEU.

Data used by the commander selection logic is moved to local data areas under interrupt protection to ensure that the data is consistent across the Redundant Set. In addition, each GPC in a Redundant Set assumes that all other GPCs in the Redundant Set have the same active memory configuration number, MF ID, and overall Redundant Set mask that it has, instead of using corresponding ICC data.

The hierarchy used for selecting a new commander is as follows:

(1) If the new major function of the DEU is supported in a Redundant set, the New Bus Commander is determined from the GPC CRT Cmdr value for the DEU. If the GPC specified in that field is not in the RS, the new commander is determined from the Nominal Bus Assignment Table. However, if the GPC specified in the table entry is not a member of the Redundant Set, the lowest GPC in the Redundant Set becomes the New Bus Commander, and the Nominal Bus Assignment Table is updated accordingly.

(2) If the new major function of the DEU is not supported in a Redundant Set but is supported by a simplex GPC, the simplex GPC becomes the New Bus Commander and the Nominal Bus Assignment Table is updated accordingly.

(3) If the new major function of the DEU is not supported by either a Redundant Set or a simplex GPC, the current commander retains command of the DEU.

If the DEU MF is active in the Common Set in both a Redundant Set and in a simplex GPC (e.g., after a Redundant Set fail-to-sync), the DEU is assigned to the Redundant Set. Or, if the DEU MF is active in more than one simplex GPC in the Common Set (e.g., after a Redundant Set fail-to-sync), the DEU will be assigned to the lowest numbered Common Set GPC with the DEU MF active.

e. GPC Reconfiguration Message Handler

This module provides the interface between the System Interface Processor and the GPC Reconfiguration module. It acts as a service processor to GPC Reconfiguration.

This module receives the ICC Message to initiate OPS transition processing via GPC Reconfiguration. The ICC data is validity checked to prevent ICC pollution. Valid ICC messages result in data initialization and a Schedule of GPC Reconfiguration.

f. Downlist Restart

This module is responsible for rescheduling the downlist I/O following the change in downlist format during GPC reconfiguration.

The time of day for rescheduling the downlist I/O is computed. Next a WAIT is issued for the time of day just computed. When the WAIT is satisfied the downlist I/O will be rescheduled.

This program executes at a priority such that it will not be delayed in rescheduling the I/O.

C. System Specialist Functions

The system Specialist Functions are the set of specialist functions (SPECs) which are system oriented as compared to being directly related to a major function. These SPECs are capable of being initiated from any major function switch setting. They are permanently resident in memory with other System Services functions because of their commonality across major functions. Figure 11.3-7 shows the breakdown of modules in System Specialist Functions.

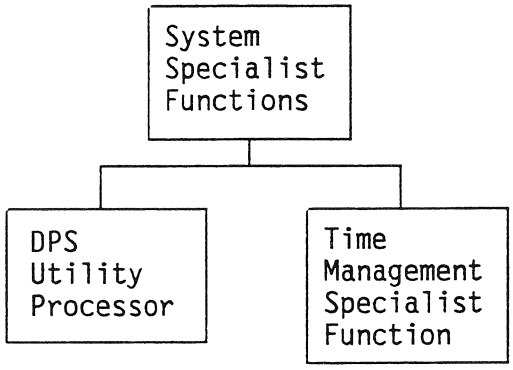


Figure 11.3-7.- System Specialist Function.

1. DPS Utility Processor

   The DPS Utility Processor consists of all the functions required to support the DPS UTILITY display SPEC 1.

2. Time Management Specialist Function

   The Time Management Specialist Function provides control of the GPC Time Management Processor and updates to the master timing unit (MTU) MET/GMT accumulators and GPC MET/GMT. Provisions are made for monitoring the MTU accumulator strings, the GPC Self Time, and time management processor source selection. Inputs are provided for initiating event timers and for Class 3 alarm annunciation of specified values of the event timers or mission times.

D. CRT Displays

The CRT Display section describes the contents of displays that are maintained by System Control rather than by the application areas. These displays are associated with the Operational Sequence and Specialist Functions that are documented under System Control.

The System Control CRT displays consist of the following:

a. GPC Memory Display (SPEC 0)

b. GPC/Bus Status (SPEC 6)

c. Time Management display (SPEC 2)

d. Fault Summary (SPEC 99)

e. DPS Utility display (SPEC 1)

E. References

1. System Software Detailed Design Specification, part 3

11.3.1 ICC BUFFER and ICC SUMWORD PROCESSING[1]

11.3.1.1 Historical Perspective

Prior to STS-1, only redundant set output sumword processing existed in the software. Subsequently, fixed and variable ICC buffer sumword processing was developed using redundant set output sumword processing as a development model. Redundant set output sumword processing produced a SUMWORD X fault message but was deleted from the software in OI-8A.

11.3.1.2 Intercomputer Communication Checksum Processing

11.3.1.2.1 General information.- The Intercomputer Communication (ICC) process is the way by which all active PASS GPCs know that each other exists and how they communicate critical systems information. The ICC process, which is coordinated by module AIESIP in the System Control area of System Software, runs at 6.25 Hz (takes four minor cycles to complete) and is tied to the SSIP timer interrupt that starts each minor cycle. See the System Control Systems Brief (11.3) for more details on AIESIP.

The AIESIP ICC processing is broken up into four phases (see System Brief 11.3 for details) and involves the physical transfer of a 124-halfword block of data called the ICC buffer. This buffer is broken down into a fixed message portion (74 halfwords) which is always in the buffer and the variable portion (50 halfwords) that will contain various messages when specific events occur (like adding a GPC to a redundant set). Relative to the fixed ICC data transfer rate, the variable ICC message buffer is used at most once per second. The fixed ICC message has most of the data a PASS GPC normally needs to belong to the Common Set (CS). Table 11.3.1-I lists the flags associated with the variable ICC data and shows how many halfwords of data are associated with each flag.

There are two types of ICC checksum processing. Checksumming is performed on the variable portion of the ICC buffer. See table 11.3.1-III for a list of the ICC buffer contents. The second checksum is performed on a set of critical system software parameters, which are listed in table 11.3.1-II.

The method of coordinating variable buffer messages in the CS is via the total length of all the variable messages to be sent, which is stored in the fixed message by the sending GPC. Also, the sending GPC sets an ICC flag (48 in all) for each variable message, so its ICC software knows which messages are to be sent on any given ICC cycle. These flags are contained in a three-halfword array, CZ2B_ICC_FLAG. This variable is not part of the fixed ICC buffer, but is only used internally by each GPC to determine what variable ICC data to send. See table 11.3.1-I for a list of the variable ICC flags. The total variable buffer length is interrogated by the receiving CS GPCs and allows for the processing of all the variable buffer messages that were sent. One to three ICC variable buffers can be sent and processed at one time. For example, the once per second annunciation

variable buffer message takes up the entire variable portion of the ICC buffer.

11.3.1.2.2 Variable ICC checksum.- With this basic understanding of the ICC process, we can now explore the world of ICC checksums. The variable portion of the buffer, if any data is being transferred in it, is checksummed every ICC cycle (160 ms). If the checksum compare fails two consecutive times, the appropriate GPC will fail-to-sync with a GPC error code of 0507 logged. Only the variable portion of the ICC buffer is checksummed in this manner because (1) the variable data is critically needed the moment it is sent, and (2) AP101B CPU overhead was too high to include processing of this type for the fixed portion. However, the AP101S is fast enough to include the processing for the fixed portion, but it is too expensive to implement.

Since a fresh copy of the fixed data is continuously sent, the risk of using erroneous data was considered small. The FCOS software module that detects a bad variable buffer checksum is FIOCMPLT.

11.3.1.2.3 SUMWORD ICC X.- The SUMWORD ICC X error message annunciation is due to an entirely different process. The reason for its existence is to cover an AP101B IOP local store register failure which has no parity protection. The AP101S does have IOP parity detection, but it is not currently enabled (see Section 3, AP101S SB).

A checksum is used to detect differences in critical system software parameters. These differences could result from pollution or timing situations. For example, when a GPC is added to the common set (CS), many ICC flags are set for initialization of the new GPC. During this ICC (which will take several ICC transmissions) if certain control messages are also sent as a result of crew input, differences could appear between CS members in data values. The system software sumword is placed in the fixed data portion of the ICC buffer which is calculated using the sum of critical system software parameters (see table 11.3.1-II). Each GPC does a check of every other GPC's sumword every four major cycles.

After three consecutive miscompares (which takes 11.52 seconds), it will annunciate a SUMWORD ICC error message against the GPC at fault or against itself if it disagrees with all other GPC's sumwords. Only one error message will be annunciated, although the miscompares will continue until the sumword matches or the GPC is reintroduced into the set (when the miscompares may start again). The FCOS software module that detects a bad sumword, increments or zeros a sumword counter, and causes annunciation of the SUMWORD ICC error message is FPMIHPC2.

The SUMWORD ICC software is modeled after GNC SUMWORD X error message processing, which was deleted in the OI-8A software release.

The system software sumword calculation is disabled (for 16 major cycles) when an RS set expansion is performed, precluding a nuisance message from being annunciated while a known large amount of data is being transferred to the new RS member(s).

11.3.1.2.4 ICC message flags.- Variable ICC messages can be built in two ways. First, the software module DMC_ICC_Collector can build a variable ICC message, or the software module identifying the need for a variable ICC message can also build the message. The following list of flags is used so the sending GPC knows which messages are to be put in the 50-halfword variable portion of the ICC buffer CZ2B_ICC_MSG_BUF.

TABLE 11.3.1-I.- VARIABLE ICC BUFFER FLAGS

Flag number	HAL name	Numbers of Halfwords
1	CZ2V_MTU_UPDATE	36
2	CZ2V_MC_REQ	34
3	CZ2V_NOM_BUS (1,1)	26
4	CZ2V_NOM_BUS (3,1)	26
5	CZ2V_NOM_BUS (5,1)	26
6	CZ2V_NOM_BUS (7,1)	26
7	CZ2V_NOM_BUS (9,1)	26
8	CZ2B_ACT_(1;)	20
9	NULL	0
10	CZ2B_IO_ERR_GPC (1,1:)	9
11	CZ2B_ERR_GPC (1,1:)	7
12	CZ2B_ACT_XMTR1 (1;)	8
13	CZ2V_CURRENT_OPS (1;1)	9
14	CZ2V_TB_ID (1)	1
15	CZ2B_CS (1;)	2
16	CZ2B_TIME_SOURCE	1
17	CZ2V_GPC_P	1
18	CZ2V_SRB_BUS_REQ	2
19	CZ2B_BTU_BITE_MM1 (1:)	7
20	CZ2B_MM_MF (1:)	4
21	CSCV_SCM_ICC (1:)	1
22	CGEV_SM_AM_DATA (1:)	40
23	CGEV_SM_SCM_DATA (1:)	42
24	CGEV_SM_AM_DATA (1:)	31
25	CZ1V_PMU_SELECT	1
26	CZ2B_HDR_LDR_IND	1
27	NULL	0
28	CZ2V_MTU_PARM_ICC_ADDR	14
29	CZ2V_TM_STAT_FLG	1
30	CZ1B_CKPT_RETRV_ENA	1
31	CZ1V_EVT_ZERO_GMT (1;)	32
32	CZ1V_EVT_ZERO_GMT (2;)	32
33	CZ1V_EVT_ZERO_GMT (3;)	32
34	CZ1V_TIME_SEL	4
35	CDUV_CURRENT_NSP	1
36	CDL_LAP_LIMIT	1
37	CDMV_UI_DOWNLIST_FORMAT_ID (1;)	1
38	CDMV_UI_DOWNLIST_FORMAT_ID (2;)	1
39	CDMV_UI_DOWNLIST_FORMAT_ID (3;)	1
40	CDMV_UI_DOWNLIST_FORMAT_ID (4;)	1
41	NULL	0
42	NULL	0
43	CZ2B_DPS_STATUS	1
44	CZ2V_MTU_UPDATE	2
45	CZ1B_D_UL_CNTL	1
46	TFUM_DEU_CMDR (1)	4
47	CZ1B_D_UL_CNTL_SM	1
48	CZ2B_ISOLATED_DEUS	1

11.3.1.2.5 SUMWORD ICC X variables.- This list of data is used to form the system software sumword that is placed in the fixed portion of the ICC buffer. Associated processing can lead to a SUMWORD ICC error message:

TABLE 11.3.1-II.- FIXED ICC BUFFER SUMWORD CALCULATION INPUTS

HAL name	Internal FCOS reference(s)
CZ2V_TSIP	CZ2VTSIP
CZ2V_MET_MSEC	CZ2VMETM
CZ2V_MET_HALFHRS	CZ2VMETM+2
CZ2B_MODE	CZ2BMODE
CZ2V_MM_MF	CZ2VMODE+1
CZ2V_SRB_BUS_REQ	CZ2VMODE+5
CZ2V_LDB_BUS_REQ	CZ2VMODE+6
CZ2V_DEU_MF	CZ2VMODE+7
CZ2B_DPS_STATUS	CZ2VMODE+11
CDUV_CURRENT_NSP	CDUVCURR
TFCM_DEU_CMDR	TFCMDEUC
CZ2V_CURRENT_OPS	TGSTPOPS, TGSTGOPS, TGSTSOPS
CZ2V_PROG_OVLY	TGSTPGOV
CZ2V_MF_OVLY	TGSTMFOV
CZ2V_NOM_BUS	CZ2VNOMB
CZ2B_ISOLATED_DEUS	CZ2BISOD
CZ2B_GRT_GPC_SET	CZ2BGRTS

11.3.1.2.6 ICC message buffer contents.- The priority of messages into the buffer are annunciation messages (SIP Cycle 0), other messages collected since last ICC, and message flags. The message flags are used to indicate what data needs to be updated via ICC. The fixed ICC buffer data are located in HAL/S structure CZ2V_ICC_BUF in compool CZ2_COMMON. This data is always passed between all common set GPCs via the 124-halfword ICC buffer at a rate of 6.25 Hz.

TABLE 11.3.1-III.- ICC MESSAGE BUFFER CONTENTS

HAL name	Function	Number of halfwords
CZ2B_OVERRUN_IND	INDICATE SIP CYCLE OVERRUN	1
CZ2B_STATUS	ICC BUFFER FLAGS (VT WD)	1
CZ2V_DUTY_CYCLE	% CPU UTILIZATION	2
CZ2V_SUM_WD	SUM WORD	2
CZ2V_SSW_SUMWORD	YIELDS SUMWORD ICC X IF MISCOMPARES OCCUR	1
CZ2V_VAR_BUF_LENGTH	LENGTH OF VARIABLE ICC MESSAGE	1
CZ2B_DIA1	DISCRETE INPUT A 1ST HWD	1
CZ2B_DIA2	DISCRETE INPUT A 2ND HWD	1
CZ2B_DIB1	DISCRETE INPUT B 1ST HWD	1
CZ2B_DIB2	DISCRETE INPUT B 2ND HWD	1
CZ2B_GPC_MT	MSEC OF GPC GMT CLOCK	2
CZ2B_GPC_MT2	1/2 HRS OF GPC GMT	1
CZ2B_DK_GSE DP	GSE & DK DATA PATH INDICATIONS	1
CZ2V_MF_DPS	MAJOR FUNCTION DISPLAY CODE	2
CZ2B_OPS_DPS	OPS DISPLAY CODE	2
CZ2B_DP_DISP	DATA PATH STATUS 1ST HWD - PRIMARY + SRB 2ND HWD - SECONDARY + MCIU	2
CZ2B_NSP_BUFFER	NSP DATA	36
CZ2B_MTU_BUFFER	TIMES USED BY FPMMTURM	14
CZ2B_ICC_ERROR_BUS_MSK	ICC ERRORS DETECTED	1
CZ2B_ICC_MSG_BUF	VARIABLE ICC MESSAGE BUFFER	50
CZ2V_ICC_CKSUM	VARIABLE ICC MESSAGE'S CHECKSUM	1
	TOTAL	124

11.3.1.3 References

1. Space Shuttle Programs Orbiter Avionics Software Operational (OBS) Detailed Design Specification, Volume: II-System services:

   Part 1: Flight Computer Operating System (FCOS), sections 3.2.4, FIOCMPLT, and 3.1.2.3, FPMIHPCZ

   Part 3: System Control (SC), section 3.1.2, AIESIP

11.3.2 PASS OPS Transitions

An OPS transition is the process used to load applications software into a general purpose computer (GPC) after initial program loading (IPL'ing) or to change the software in a previously loaded GPC. The former is obvious; a GPC must be loaded with software to be useful. However, the latter is not as obvious. Because the shuttle computers must handle the operational environments posed by ascent, orbit, entry, payload operations, and systems management (SM), a large amount of code is required. This amount of code (564K) exceeds the amount of memory space available in the expanded memory of the new GPC (AP101S, 256K). Therefore, a system of software division, organization, and transfer was developed to allow for an organized transition from the use of one set of software applications to another. This is called reconfiguration and involves a process of loading successive grouping of software into GPC memory. Because each successive load may overlay some of the code already present, these software loads are called overlays. The entire procedure for stopping the execution of one group of software and handing over to the use of another is known as an OPS transition.

The OPS transition is a complex process that involves from one to four computers, software stored in multiple locations including a relatively slow electromechanical device (tape drive) called a mass memory unit (MMU), and software transfer over multiple data buses. This all requires a complex set of software applications for proper control of the process.

11.3.2.1 Memory Configuration Software Organization

To understand the OPS transition process requires that the organization and division of the flight software be understood.

A. Software copies

At the highest level, there are three identical copies of all flight software on each MMU (except for BFS and main engine controller starting with OI-21). These copies provide a high level of redundancy for protection against MMU failures or tape area problems on a given MMU and are identified as copy 1, 2, and 3 for each MMU. Controls are provided for selection of the copy to be used (sec. 11.3.2.2.B).

B. Phases

Within each copy, a major software area is identified with a phase number. These phase numbers are identified in a table in the Mass Memory Unit Program Release Notice (MMU PRN), JSC-161764. A sample phase assignment table is shown in table 11.3.2-I. Note that the table makes reference to "memory configurations" (MCs). Because the software required to support any one part of the mission requires a combination of system software (SSW), major function base (MFB), and OPS overlays, each particular combination of software is designated with a memory configuration number. There are nine MCs currently defined, although the software MC table (CZ2V-GRT-TAB) is set up for a total of ten

configurations. See table 11.3.2-II for a sample configuration table. Notice that for each configuration the table provides the phase number of the primary OPS overlay and the MFB overlay, the MF number of the configuration, the OPS number used to identify the configuration, and the maximum number of modes available within that configuration.

TABLE 11.3.2-I.- MASS MEMORY PHASE ASSIGNMENTS.

Phase no.	Memory config	Description	Dec no. halfwords
1*	N/A	GPC bootstrap loader and mini-directory	3304
2	ALL	Resident system software	104750
3	1,2,3,8,9	GNC major function base (GNC-FB)	9656
4	1	Ascent and orbit OPS (GNC-1/6-A)	16776
5	2	On-orbit OPS (GNC 2-A)	27660
6	3	Entry OPS (GNC 8)	22334
7	8	VU on-orbit (GNC 8)	124354
8	9	VU pre-count (GNC 9)	122170
9	6	VU checkout function base (VCO-FB)	8346
10	1	Ascent and abort OPS (GNC 1/6-B)	116574
11	3	Entry OPS (GNC 3-B)	111036
12	6	VU mass memory OPS (PL9)	100850
13	2	On-orbit OPS (GNC 2-B)	105586
14	4,5	SM function base (SM-FB)	27290
15	4	Orbit-doors OPS (SM2)	114996
16	5	Spacelab support OPS (SM4); reserved	114992
17,18	N/A	Reserved for growth	
19	N/A	Payload control supervisor sequences (PCS)	
20	N/A	P/L checkpoint; reserved	
21	N/A	SM data checkpoint	
22	N/A	IMU calibration checkpoint	
23	N/A	PDI decom format load (DFL)	
24	N/A	PDI fetch pointer memory load (FPL)	
25	N/A	Telemetry format load (TFL)	
26	N/A	GPC-STP/SW loaders (GPC-STP, SSL, LOAD TBL)	
27-34	N/A	Subsystem configuration management (SCM)	
35-60	N/A	Roll-in-displays	

*Not currently in the PASS phase table

TABLE 11.3.2-II.- GPC MEMORY RECONFIGURATION TABLE (CZ2V_GRT_TAB).

	MC	OPS Ph	MF Ph	MF value	OPS	Modes
G1A	1	4	3	2	1	6
G2A	2	5	3	2	2	2
G3A	3	6	3	2	3	5
S2	4	F	E	3	2	2
SM4	0	10	E	3	4	2
PL9	6	C	9	1	9	1
	0	D	9	3	9	1
G8	8	7	3	2	8	1
G9	9	8	3	2	9	1
	0	0	0	0	0	0

1 = PL
2 = GNC
3 = SM

C. Loadblocks

Each phase of software is further subdivided into loadblocks. Each loadblock represents a grouping of programs that all reside in a contiguous area of GPC memory. The mass memory build microfiche delivered with each software release lists each phase, the loadblocks it contains, and the control sections (CSECTs) located in each loadblock. Note that a loadblock may contain from 1 to 256 MM blocks. The MMU address for each loadblock is provided as is the GPC address for each CSECT. CSECTs are individual programs, data compools, or patch areas.

D. FCOS vs. applications

All of the software programs are either flight computer operating system (FCOS) programs or applications programs. The FCOS could be compared to the disk operating system (DOS) in your PC, and the applications to all the "user" programs.

The FCOS programs handle the low-level tasks like the actual BCE programs that "talk" to the MMU or programs that enable GPC-to-GPC transfers. The FCOS programs also handle the actual reconfiguration of GPC memory resulting from OPS transitions and the error detection routines. All FCOS program names are six letters in length starting with "F." All FCOS programs are written in high-level assembly language (HLAL) for absolute control over hardware functions, thereby providing precise timing of processes and efficiency of code.

Applications programs are higher level programs. Typical examples are the programs that are used to process keystrokes, process OPS requests, locate needed software, initiate overlay processes, and call on FCOS programs via supervisory calls (SVCs) to do the detail work of interfacing with MMUs and moving software. Application program names are combinations of words or acronyms joined by underscores "_" and are written in the HAL language.

E. Software types

As discussed previously, all of the software is being written in either HAL (applications) or in assembly language (FCOS). Another way to classify the software is by the tasks it is designed to handle. Using this division, there are basically five different types of software: IPL, SSW, MFB, OPS, and other special applications/data. The order of software loading into the GPC is always the same. IPL gets the GPC started; then SSW is loaded, followed by MFB; and, finally, the OPS software is loaded. Other special data may be loaded, as required from the MMUs. Each load may overlay or replace part of the software previously loaded, which enables the GPC to support more specialized operations with each additional layer of software added. OPS transitions involve loading of OPS and MFB overlays.

The IPL software is loaded into the GPC by the bootstrap program, runs the self-test program to determine the health of the GPC, and then loads the SSW. (See section 9.1 for details of the IPL process.)

The SSW is loaded by the IPL process and remains in the GPC until it fails or a new IPL is performed. The SSW takes care of background tasks that allow a GPC to run and communicate with the outside world over data buses and respond to certain hardwired control switches and signals. The SSW enables multiple GPCs to operate together in common sets (CS) or redundant sets (RS). Having an RS provides redundancy in software operation and vehicle control through each GPCs use of identical inputs to identical processes and to generate identical commands at the same time. SSW also contains the OPS transition programs. SSW comprises the FCOS and applications programs.

The MFB software is one of three types; guidance navigation and control (GNC), systems maintenance, or payloads support (PL). GNC software supports vehicle control during ground checkout and all flight phases. Systems maintenance is used to monitor and control vehicle support systems, interface with payloads, and control payload bay doors. PL MFB was originally designed to support payloads but is now more of a special purpose software that supports large MMU patches and display electronic unit (DEU) loads. MFB software is all applications type programs.

The OPS software is specialized to support a particular part of flight. There is different OPS software to support prelaunch, ascent, orbit, entry, and postlanding phases of flight. OPS software is all applications type programs.

The other/data type software remains on the MMU and is called for as needed. Examples are telemetry format load software for the pulse code master modulation unit (PCMMU), roll-in displays for displays needed only occasionally, main engine controller data loaded prelaunch, and inertial measurement unit (IMU) calibration data stored prelaunch.

11.3.2.2 Philosophy of OPS Transitions

Before investigating the actual software used to control an OPS transition, it is important to understood these concepts: transition types, software sources, transfer methods, GPC(s) involved, and assurance of successfully completing a transition.

A. Transition types

OPS transitions can be of four different types: OPS 0, non-OPS 0, mode recall, and mode request. Of these four, only the non-OPS 0 request requires that different software be loaded into the GPC. The others use existing software in the GPC. The OPS 0, non-OPS 0, and mode recall are all initiated by keyboard entries. The mode request can be made by keyboard entry, by automatic software sequencing, or by abort selection.

The freeze dry (FD) request, while not officially an OPS transition, nevertheless is a reconfiguration of GPC memory that warrants discussion here. In fact, it does utilize some of the same code used by the OPS transitions.

The OPS 0 type of transition is used to cause an orderly termination of the programs currently running. This transition requires no additional software to be loaded, but does require that shutdown and cleanup routines be run.

The non-OPS 0 type of request covers the cases where a different set of software is needed to perform the particular function requested (i.e., control the orbiter during the entry phase).

A mode recall is a request for the same software that is currently running. This transition is done to cause a reassignment of bus commanders, or in certain ops, this can be done to change the target set of GPCs (also a reconfiguration, but in a different sense).

A mode request causes a different part of the resident software to become active. An example of this transition occurs during ascent when there are several OPS 1 modes used to divide up the different phases of ascent. A mode request can be initiated by a keyboard entry, or it can be caused automatically by software sequencing or by the selection of an abort mode. (RTLS causes selection of 601, which is a mode existing within the OPS 1 software.)

An FD request is a very special memory reconfiguration initiated by GPC MEMORY item entries rather than the OPS XXX PRO used for standard OPS transitions. The FD request item entries are valid only for a GPC in OPS 0. The entries cause a selected memory configuration to be loaded into memory but then leaves the GPC in an OPS 0 status. (The newly loaded configuration is not initiated.) Prior to OI-20 and the introduction of G3 archive, the FD GPC was always loaded with entry software to provide assurance of having a GPC with entry software (G3) in case both MMUs fail while on orbit. With G3 archive and the entry software present in the upper core of all PASS GPCs, the FD GPC is now loaded with orbit software (G2). An FD GPC provides a quick way to recover vehicle control in the event of loss of the current GNC GPCs.

B. Sources of required software

Transitions that require different software; i.e., a non-OPS 0 request, must have a source for the required software. There are two places to obtain the software, the MMU or another GPC. Software can be transferred via MMU data buses or the launch data buses (LDBs).

For redundancy, the orbiter uses two identical MMUs, each with three software copies and each served by an MMU bus (1 and 2, respectively). The software copy utilized is determined during the IPL process (IPL MENU item entry) or via item entries on the GPC MEMORY SPEC (items 52-54). The IPL copy selection causes all subsequent software

use from the MMU to come from that copy. The GPC MEMORY items allow copy selection per MF, but only for software associated with OPS transitions for that particular MFB. This also drives the software copy used for the G3 archive load prelaunch at the G9-G1 transition. All software other than OPS transition software will come from the copy selected at IPL.

Another GPC can be used as a source if it contains the required software. The GPC can have the software in low core, or, in the case of the G3 software, it could be in upper core as the G3 archive. Transfer can occur over the MMU buses or over the LDB buses.

A previously available source of entry software involved uplinking it from the ground (entry memory uplink (EMU)). Uplinking software was a very special technique developed to assure a source of entry software in case the onboard MMUs both failed. After a GPC failed a G3 FD attempt, then a special enable function allowed the GPC to receive the uplinked entry software. With the advent of the G3 archive function introduced with the OI-20 software release, the need for the EMU went away. Although the onboard software capability remains to accept the EMU, the ground processing to build the EMU is no longer supported. (STS-43 was the last flight of full EMU support.)

C. Selecting sources of overlays

Source selection for the new software involves selection of source for the MFB overlay and a source for the OPS overlays. When the source is an MMU, there is little confusion since both the MFB and the OPS come from the MMU. However, when the source is a GPC, the logic is not as clear. It is possible for the MFB overlay to come from one source and the OPS overlay from another (GPC and MMU or GPC and another GPC).

The first question to answer is whether the MMUs or a GPC will be used as a source (MFB or OPS). The MMUs will be used as the overlay source if MMU/MMU (item 10 on DPS UTILITY) is selected on the DPS UTILITY SPEC or if GPC/MMU (item 9 on DPS UTILITY) is selected and no GPC has the required overlay. For the case where GPC/LDB (item 11 on DPS UTILITY), GPC/MMU, or G3 archive retrieve (item 49 on DPS UTILITY) is selected, the source GPC(s) must be determined.

The concept of primary and secondary GPCs, as defined by the reconfiguration process, is the key to determining a source GPC. A primary GPC is a GPC receiving the reconfiguration request (commander of the CRT receiving the request) or any GPC in a redundant set with this GPC. Secondary GPCs are any other GPCs targeted for the transition. In addition, these primary and secondary GPCs must meet the qualification of either being in the current redundant set, the current common set, and in ops 0, or being the GPC receiving the reconfiguration request.

The FD operation is an exception to these rules for picking GPC or MMU as an overlay source. An FD request will cause the MMU to be used as an MFB overlay source if the targeted GPC does not have the required MFB

and will also use the MMU as the source for the OPS overlay. Note that the OPS overlay is always done for an FD request even if the GPC already has the requested OPS overlay.

For other than the FD request, once the primary and secondary GPCs have been determined, the logic builds mask words identifying each targeted GPC containing the required MFB and OPS and also sets the inverse bits identifying GPCs without the required overlays. The logic identifies the lowest ID GPC in each of four categories, primary GPC with MFB, primary GPC with OPS, secondary GPC with MFB, and secondary GPC with OPS. In most cases all of these categories will not be filled. If there is a primary GPC with MFB, as selected above, it is set as the MFB source; otherwise, the secondary GPC source is selected. The same priority is used to select the OPS source. If a G3 archive retrieve is selected and a G3 transition is being requested, then the targeted GPCs are examined for the presence of a G3 archive load. If not all targeted GPCs have the archive, then the lowest ID-targeted GPC becomes the source for the overlay. If no GPCs have G3 archive present, then the MMU becomes the source for the overlays.

It should be evident from these discussions that there is no way that the software distinguishes "old" MFBs from "current" MFBs. The MFB used is determined entirely by the priorities and rules discussed above. Therefore, procedural care must be exercised in performing OPS transitions to ensure the desired MFB is obtained. This is why making the OPS call to the FD GPC when adding it to the set is the wrong thing to do. The old MFB in the FD GPC would be transferred to other GPCs with the current MFB, thereby trashing the current vehicle status information stored in the current MFB.

D. Selecting destination GPCs

The selection of destination GPCs also utilizes the primary/secondary GPC concept. If the overlay source is a secondary GPC, then all other target GPCs receive the overlay. If the source is a primary GPC, then all secondary GPCs and any primary GPCs without the overlay will receive the overlay. This process is first used to determine MFB overlay destination GPCs and then is used to determine OPS overlay destination GPCs.

E. Assurance of transition completion

Because OPS transitions involve a suspension of active vehicle control, OPS transitions are designed to minimize the time of no control and to make the best attempt at successfully loading and running the software for the new OPS. Certain features are built into the logic expressly for providing some assurance of successful OPS transitions. This assurance is provided by pre-positioning of the MMUs, redundancy managing the MMU ready/busy discretes, and checksumming the software transfers.

1. Pre-positioning

   Pre-positioning of MMUs is utilized in OPS transitions that require software from the MMUs (except for FDs). Pre-positioning tests the GPC-MMU interface prior to committing to the transition. The test sends a command to each MMU to position the tape at the location of the required software. This transaction includes a request for the MMU internal BITE words (MMU status words). The logic confirms the existence of a good interface through the receipt of the BITE words from the MMU. This operation is called a pre-positioning because later, when the actual request goes out to the MMU to read in the required software, another position tape command is sent to the MMU before the actual read command. Keep in mind that pre-positioning is only associated with OPS transitions. The GPC MEMORY FD controls initiate a very special type of reconfiguration process that loads the software but does not initiate the software. Therefore, there is no pre-positioning associated with the FD process because there is no concern about having a period of no vehicle control as there is with a standard transition.

2. Retry and toggle

   The transition logic incorporates a retry on MMU transactions and toggle logic to try both MMUs, if necessary, to get a good transfer of software to all targeted GPCs. A retry is built into the transition logic because the MMUs are an electromechanical device, providing protection against transient MMU read problems or any disturbance on the data bus that might affect the transferred data. Normally, a transition will only need to access a single MMU to obtain the needed software. If there is a problem; i.e., the MMU cannot deliver or one of the targeted GPCs has an MMU bus receiver problem, then the transition logic is designed to try the alternate MMU and its associated bus. This retry and toggle logic also is applied to GPC-to-GPC (GTG) transitions and FD loading. Instead of trying the other MMU for GTG, the change involves using the alternate MMU (or LDB if selected) bus and trying again to transfer the software.

   A checksum is used on each section of data (designated as a loadblock) to provide an integrity check on the transferred software. Each loadblock stored on the MMU has a checksum as the last word in the loadblock. A new checksum is generated as the data is transferred into a GPC and compared with the received checksum. If they do not agree, then the transaction is flagged as bad, an I/O error is logged, and that data is not used. This procedure is used also for GTG transfers at transitions.

3. Ready/busy and IPL select discretes

There are two discretes associated with each MMU that affect the ability to properly use an MMU. These are the MMU ready/busy discrete and the MMU selected for IPL discretes. The ready/busy discrete is generated within each MMU and hardwired to each GPC. When set to busy, this indicates an MMU is receiving a command, processing a command, transmitting data, or is not powered. Otherwise, this will indicate ready. Because these discretes are hardwired to each GPC, a receiver failure at a GPC could cause that GPC to sense the wrong status for an MMU. Similarly, the MMU IPL SOURCE SELECT switch on panel O6 is hardwired to each GPC to indicate that a particular MMU has been selected to support an IPL. If an MMU is selected for IPL use, then it is not available to support an OPS transition. To protect against a GPC discrete receiver failure, a process of redundancy management (RM) is used on these individual discretes. The software for MMU access utilizes a single RM'd discrete rather than all the individual GPC discretes. The RM logic must see a majority of the common set GPCs indicating a change in the status of the discrete before the RM'd discrete is changed. This means that if there are four GPCs in the common set, then three GPCs must see a change in the discrete before the RM'd discrete is changed.

A single switch on panel O6 is used to select either MMU for IPL. Therefore, if this switch had an internal failure causing both MMUs to be selected for IPL, there would be no MMUs available to support a TAL abort OPS transition. Because this was the most critical OPS transition prior to the introduction of the G3 archive, a change was made starting with the OI-8D software to provide a means of masking these IPL select discretes in order to protect the TAL transition. The control appears as an item 38 entry on the DPS UTILITY SPEC and is called IPL SOURCE SW MASK. When set, this mask causes the software to ignore the MMU IPL discretes in determining the availability of the MMU to any non-IPL transaction. The mask is set by entering the item 38 on DPS UTILITY and is also set automatically at the prelaunch G1 software initialization. Even though the G3 archive is available to support the TAL transition, the mask capability remains. The IPL source select discretes are RM'd in the same way that the ready/busy discretes are RM'd.

11.3.2.3 OPS Transition Control

There are multiple controls over transitions, some coming from keyboard entries and some coming from built-in control logic or tables.

A. Transition table

One control is the transition table, known by its HAL name of DM6V_TR_TAB. This table is also called a Legal OPS Transitions table and is included in the Level A (SS-P-0002-170) as table 6-2B.

In software this table is just an array of full words with no particular meaning. When you spread out the table in the form of the level A table, as shown in table 11.3.2-III, then its use becomes apparent. It provides the software a method to determine which OPS transitions are allowed based on a "from/to" arrangement.

TABLE 11.3.2-III.- TRANSITION TABLE (DM6V_TR_TAB).

	FROM \ TO	000	101	102	103	104	105	106	201	202	301	302	303	304	305	601	602	603	801	901	201	202	901
	000	1						1					1							1	1	1	1	
	101																						
	102																						
	103					1																	
	104						1						1										
	105							1					1										
	106								1				1										
	201								1	1	1												
	202								1														
GNC	301								1			1	1										
	302										1			1									
	303										1				1								
	304															1							
	305																			1			
	601																1			1			
	602																	1					
	603																			1			
	801							1			1		1										
	901	1	1																	1			
SM	201	1																			1	1	
	202	1																			1	1	
PL	901	1																					

Note: Restring available on recall of any ops.

B. MC/phase table

The MC/phase table (CZ2V_GRT_MC_PHASES) shown in table 11.3.2-IV relates the individual phases to each MC. Since the GPC will know what MC is loaded in itself (and other GPCs), by using the phase table it can determine what phases are memory resident and what phases are required when a new memory configuration is requested. Because a phase can contain no more than one MMU file's worth of software, a given MC may require two phases of OPS overlay software. These are shown as the OV1 and OV2 columns (overlay 1 and overlay 2) in the phase table. Starting with OI-22 this logic will get revised to allow for larger phases that will hold more than a single MMU file.

TABLE 11.3.2-IV.- MC/PHASE TABLE.

(CZ2V_GRT_MC_PHASES)

MC	OPS	MFB	OV1	OV2
1	G1	3	4	A
2	G2	3	5	D
3	G3	3	6	B
4	S2	E	F	0
5	S4	E	10	0
6	PL9	9	C	0
7				
8	G8	3	7	0
9	G9	3	8	0
10				

C. SW copy selection

There are two controls used in selecting software copies; the IPL process, and item entries on GPC MEMORY for each MF (starting with OI-8D). When an IPL is performed, any of the three copies of software may be selected through the use of item entries on the IPL MENU display. (Copy 1 is used by a default IPL and is normally used). After an IPL is performed from one of the available copies, then with one exception all subsequent software retrieved from the MMU comes from that copy area. The exception is when an MM AREA entry has been made on GPC MEMORY (items 52, 53, and 54). If an item entry is made to select a particular copy for one of the MFs, then any subsequent OPS transition for that MF, requiring software from the MMU(s), uses software from the specified copy. Keep in mind that these selections are redundant-set-shared, but not common-set-shared. Execution of these item entries cause changes to MMU addresses stored in the in-core phase table.

D. In-core phase table

An in-core phase table (#PFCMGPT), stored in GPC memory as part of SSW, is used to map phase numbers into MMU addresses and GPC addresses (fig. 11.3.2-1). Note that the table only covers phases 3-16, because they are the only phases associated with OPS transitions. Phase 1 is the bootstrap used at IPL, and phase 2 is the SSW only loaded by the IPL process. All phases above 16 are other auxiliary software (nontransition). Each of the three SSW copies stored on the MMU contains a phase table containing MMU addresses for that software copy of each phase. If IPL is performed using copy 2 software, then IPL loads copy 2 SSW which includes an in-core phase table with copy 2 addresses for all MMU transitions.

#PFCMGPT (loaded at IPL)

```
01C824-01C85B  #PFCMGPT+0000    FCMMGPT
01C824-01C827     +++ COPY 1 OF 14 +++
01C824         #PFCMGPT+0000    FCMMGPT_DISP_TO_LOAD_BLK      *0038      56
01C825         #PFCMGPT+0001    FCMMGPT_NUM_LOAD_BLKS         *0007      7
01C826         #PFCMGPT+0002    FCMMGPT_STARTING_MM_ADD       *29A2      10626
01C827         #PFCMGPT+0003    FCMMGPT_NUM_CONT_MM_BLKS      *0017      23
01C828-01C82B     +++ COPY 2 OF 14 +++
01C828         #PFCMGPT+0004    FCMMGPT_DISP_TO_LOAD_BLK      *004D      77
01C829         #PFCMGPT+0005    FCMMGPT_NUM_LOAD_BLKS         *0002      2
01C82A         #PFCMGPT+0006    FCMMGPT_STARTING_MM_ADD       *2CC0      11456
01C82B         #PFCMGPT+0007    FCMMGPT_NUM_CONT_MM_BLKS      *0022      34
01C82C-01C82F     +++ COPY 3 OF 14 +++
01C82C         #PFCMGPT+0008    FCMMGPT_DISP_TO_LOAD_BLK      *0053      83
01C82D         #PFCMGPT+0009    FCMMGPT_NUM_LOAD_BLKS         *0004      4
01C82E         #PFCMGPT+000A    FCMMGPT_STARTING_MM_ADD       *2884
01C82F         #PFCMGPT+000B    FCMMGPT_NUM_CONT_MM_BLKS      *0038
01C830-01C833     +++ COPY 4 OF 14 +++
01C830         #PFCMGPT+000C    FCMMGPT_DISP_TO_LOAD_BLK      *005F
01C831         #PFCMGPT+000D    FCMMGPT_NUM_LOAD_BLKS         *0003
01C832         #PFCMGPT+000E    FCMMGPT_STARTING_MM_ADD       *29C0
01C833         #PFCMGPT+000F    FCMMGPT_NUM_CONT_MM_BLKS      *002D
01C834-01C837     +++ COPY 5 OF 14 +++
01C834         #PFCMGPT+0010    FCMMGPT_DISP_TO_LOAD_BLK      *0068
01C835         #PFCMGPT+0011    FCMMGPT_NUM_LOAD_BLKS         *0019
01C836         #PFCMGPT+0012    FCMMGPT_STARTING_MM_ADD       *2A00
01C837         #PFCMGPT+0013    FCMMGPT_NUM_CONT_MM_BLKS      *0100
01C838-01C83B     +++ COPY 6 OF 14 +++
01C838         #PFCMGPT+0014    FCMMGPT_DISP_TO_LOAD_BLK      *00B3
```

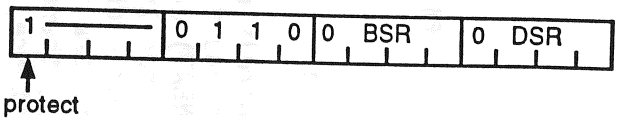
Copies cover phase 3-16

- 0038_H = Index into #PFCMGPT for start of phase 3 entries
- 0007_H = No. load blocks in phase 3
- 02982_H = MMU address for start of phase 3
- 0017_H = No. MMU blocks for phase 3 (1 MMU block = 512 words)

Figure 11.3.2-1.- In-core phase table.

```
01C854-01C857     +++ COPY 13 OF 14 +++
01C854          #PFCMGPT+0030   FCMMGPT_DISP_TO_LOAD_BLK        *0278                632                    SP INTEGER
01C855          #PFCMGPT+0031   FCMMGPT_NUM_LOAD_BLKS           *001E                30                     SP INTEGER
01C856          #PFCMGPT+0032   FCMMGPT_STARTING_MM_ADD         *1C00                7168                   SP INTEGER
01C857          #PFCMGPT+0033   FCMMGPT_NUM_CONT_MM_BLKS        *00F0                240                    SP INTEGER
01C858-01C85B     +++ COPY 14 OF 14 +++
01C858          #PFCMGPT+0034   FCMMGPT_DISP_TO_LOAD_BLK        *02D2                722                    SP INTEGER
01C859          #PFCMGPT+0035   FCMMGPT_NUM_LOAD_BLKS           *001F                31                     SP INTEGER
01C85A          #PFCMGPT+0036   FCMMGPT_STARTING_MM_ADD         *1D00                7424                   SP INTEGER
01C85B          #PFCMGPT+0037   FCMMGPT_NUM_CONT_MM_BLKS        *00EF                239                    SP INTEGER
                  +++ END OF STRUCTURE +++
01C85C-01CBA3   #PFCMGPT+0038   FCMMGPT_LOAD_BLKS                                                        ARRAY  SP INTEGER
01C85C-01C86B   #PFCMGPT+0038     *0232 *8600 *0014 *3C0A *0600 *054A *4A3A *0600 *15B4 *BCA6 *8620 *00EC *CEE0 *8630 *01
01C86C-01C87B   #PFCMGPT+0048     *8630 *098C *89BC *0650 *0006 *9D88 *0610 *34D2 *89C2 *0650 *0CF8 *9D88 *0610 *3B7A *F
01C87C-01C88B   #PFCMGPT+0058     *0136 *89C2 *0650 *2B90 *8478 *0660 *0502 *9D88 *0610 *3BAC *89C2 *0650 *174A *8478 *
01C88C-01C89B   #PFCMGPT+0068     *0246 *8600 *0154 *05E8 *0600 *001A *0618 *8600 *0052 *4154 *0600 *00F2 *4248 *0600
01C89C-01C8AB   #PFCMGPT+0078     *0600 *008C *5FEE *0600 *2012 *9D14 *0610 *2020 *CAAE *0610 *3552 *BD9A *8620 *0422
01C8AC-01C8BB   #PFCMGPT+0088     *3E44 *D00A *8630 *04BE *D4CE *0630 *07A4 *DC72 *8630 *0E84 *F486 *8630 *0B7A *A3F
01C8BC-01C8CB   #PFCMGPT+0098     *A520 *8640 *3FBE *E4DE *8640 *153C *FD7A *8640 *0286 *89C2 *0650 *2C82 *B644 *2
01C8CC-01C8DB   #PFCMGPT+00A8     *8650 *0036 *C000 *8650 *4000 *8478 *0660 *0502 *99D6 *8660 *262A *0246 *8600
01C8DC-01C8EB   #PFCMGPT+00B8     *001A *0618 *8600 *0052 *4154 *0600 *00F2 *4248 *0600 *0764 *49AE *0600 *008C
01C8EC-01C8FB   #PFCMGPT+00C8     *9D14 *0610 *0060 *9D74 *8610 *0006 *9D7A *0610 *0004 *9D88 *0610 *0350 *A0F
01C8FC-01C90B   #PFCMGPT+00D8     *0610 *00C0 *A2B4 *0610 *2954 *CC08 *0610 *2DC6 *F9CE *0610 *0632 *BD9A *F
01C90C-01C91B   #PFCMGPT+00E8     *3E28 *D00A *8630 *04C4 *D4CE *0630 *080E *DCDC *8630 *0640 *E32C *0630
01C91C-01C92B   #PFCMGPT+00F8     *A3EA *8640 *1A6E *BE58 *0640 *100C *CE64 *8640 *319C *89C2 *0650 *343C
01C92C-01C93B   #PFCMGPT+0108     *8650 *3F68 *FF68 *8650 *0098 *8478 *0660 *0502 *897A *8660 *0558 *BF
01C93C-01C94B   #PFCMGPT+0118     *052C *4A3A *0600 *002C *9D14 *0610 *005C *9D88 *0610 *0350 *A370
01C94C-01C95B   #PFCMGPT+0128     *FFF4 *8620 *000C *CEE0 *8630 *002C *D00A *8630 *02C6 *D4CE *063
01C95C-01C96B   #PFCMGPT+0138     *8630 *0122 *89BC *0650 *0006 *0246 *8600 *0234 *05E8 *0600 *0
01C96C-01C97B   #PFCMGPT+0148     *08E4 *5FEE *0600 *2012 *9D14 *0610 *0074 *D25A *0610 *03EE
01C97C-01C98B   #PFCMGPT+0158     *BD92 *8620 *3FFC *FD8E *8620 *0272 *D00A *8630 *04C2 *D4
```

- 0232_H Starting GPC address for 1st word of phase 3 load block 1
- 8600_H



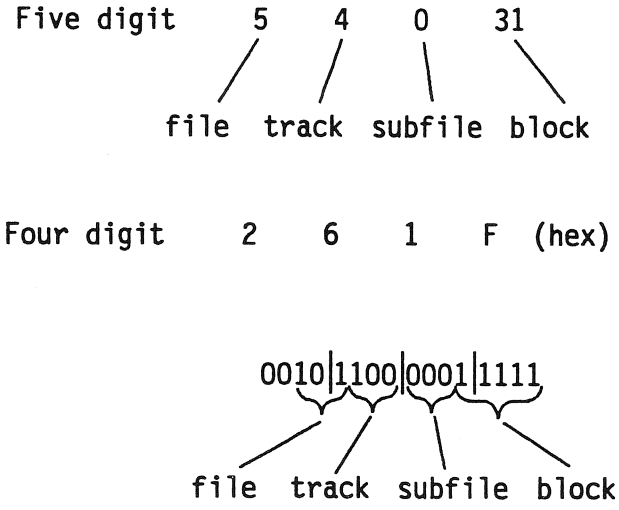
- 0014_H No. halfwords this loadblock

BCA6 8620 01EC - Phase 3 loadblock 4

- Add = BCA6 + BSR(2) = 13CA6 (See appendix H of SCP's)
- EC_H halfwords

Figure 11.3.2-1.- Concluded.

It should be noted that the MMU addresses stored in the phase table are in a compressed four-digit format rather than the familiar five-digit expression for file, track, subfile, and block.



E. Source and bus control

There are three controls over these functions. They are the MMU/source/bus controls and MMU ASSIGN on DPS UTILITY and the nominal bus assignment table (NBAT) entries on GPC MEMORY. There are three possible sources of software for transitions, the MMUs, G3 archive, and other GPCs. Because the G3 archive is GPC resident, this source reduces to a GPC source and therefore the following concentrates on GPC and MMU as sources. Similarly, there are two possible data paths available for the transfer of the software, the MMU buses and the LDB buses. Therefore, it is necessary to either select the source and data path or to provide some kind of scheme for the reconfiguration program (ARC_GPC_REC) to use in selecting them.

1. MMU/source bus

One control is provided by the MMU source/bus input option items on the DPS UTILITY spec. Item 10 (MMU/MMU) is a forcing input, which leaves the software no choice. For this input, only the MMUs are used as the source, and both the MFB overlay and the OPS overlay are read from the MMU and loaded into all targeted GPCs, whether needed or not. Using this option, current vehicle data stored in the MFB would be lost during an OPS transition. Another choice is the GPC/LDB option (item 11). This input specifies that GPC(s) be used as the source and that the LDBs be used as the data buses. If both the required MFB and OPS are not available from any of the GPCs (it could be one GPC for MFB and another GPC for the OPS), then the OPS request would be rejected and an illegal entry annunciated.

The third and most regularly used option (and initialized option) is the GPC/MMU option (item 9). This option provides the most flexibility. It causes the MMU buses to be used as the data path. The GPCs will be used as a source if they have the required MFB or OPS. If not, the MMUs are used as the source. This is the default option which is selected at system initialization and is reselected at the end of any OPS transition process if MMU/MMU was previously selected. The GPC/LDB selection is not reset to the default after the transition.

2. Bus 1 (MMU 1 or LDB 1) vs. bus 2 (MMU 2 or LDB 2)

   Beyond selecting which type of bus to use (LDB vs. MMU), there is the question of which of the two buses of each kind to use. Further, who will be in command of each bus? Bus 1 or 2 selection is determined by the MMU MF assignment to each major function by items 1-8 on DPS UTILITY. (Only one of the MMUs can be selected for each major function.) The MMU (and its dedicated bus) selected becomes the primary MMU for that OPS and the nonselected MMU becomes the backup unit. (FD loading uses OPS 0 assignments). The commander for the selected bus is set by the NBAT for the MC. If the NBAT-specified commander is not in the participating set of GPCs, then the lowest ID GPC of the set will be selected as the bus commander. (User NBAT errors will not prevent the transition from proceeding). If the LDB buses are selected, they are equated to the MMU buses to determine which one is the primary bus. (If MMU 1 is primary, then LDB 1 is primary.) Also, if LDBs are being used, the source GPC for the overlay becomes the LDB commander for the operation. After the transition is over, the LDB commander reverts back to the GPC specified in the NBAT. For GTG transitions using the MMU buses, the source GPC will be the commander of the selected MMU bus.

11.3.2.4 Overall Transition Flow

After looking at bits and pieces of the OPS transition puzzle, let us put them together and summarize the whole process. As previously discussed, OPS transitions may include the loading of the OPS overlay only, or it may include both the MFB overlay and the OPS overlay. GPCs may be added or dropped from the set during a transition and new bus commanders may be assigned. Because all of this reconfiguration means a lack of computer output to control the vehicle, an important consideration in the design of the software was to minimize the downtime between old and new OPS. However, you also want to try all the ways possible to establish the new OPS once you have taken down the existing OPS. Any GPC, which cannot successfully complete the loading of new software without errors after all attempts, will fail to synch out of the target set before the new OPS is initiated. OPS transitions are basically handled by two software programs.

DMC-SUPER processes the keyboard OPS input request and decides if a reconfiguration is necessary. ARC_GPC_REC performs the reconfiguration of GPCs by obtaining the software from the computed (or directed) source and then delivering it to the target GPC(s) using the proper data path.

For a top level walkthrough of the reconfiguration process, refer to the Big Picture flowchart (fig. 11.3.2-2), which shows the basic application modules that are used to control the overall process. Note that some modules appear more than once in the flow. Also note that SVC calls are made to FCOS programs to handle pre-positioning and to handle actual overlay acquisition.

A. Specific logic flows

To get a better picture of what activities occur for each type of transition, let us consider sample flows for set expansions, a new OPS request, and an FD request.

1. Set expansions

- Verify valid keystrokes
- Verify no transition in progress
- Verify validity of request using Valid Transition Table (DM6V_TR_TAB)
  - ok if in table or is a mode recall
- Determine target GPCs and current run GPCs
  - checks table set by GPC memory and NBAT target entries (CZ2B_GRT_GPC_SET)
  - must be in CS, in OPS 0, and targeted
  - determine if target set different than current set, if different, the transition must have been allowed by valid transition table, if same set, do bus reconfiguration
- Take set to OPS 0 (DM2_REC_PREP)
- Schedule OPS (DM5_NEW_CSEG)

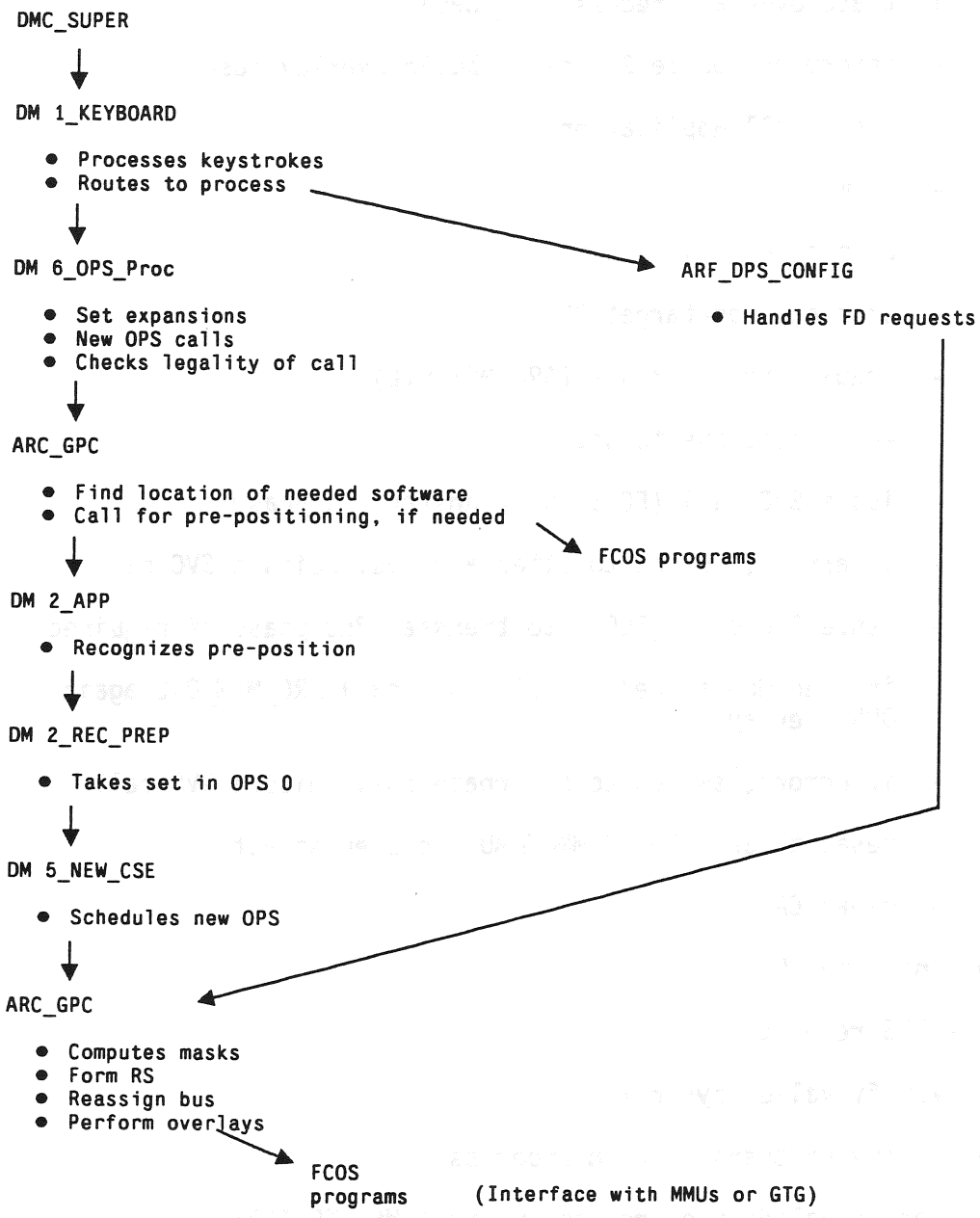


Figure 11.3.2-2.- Big Picture flowchart.

- Initiate overlay process (ARC_GPC)
  - determine source GPC(s) to build overlay masks
  - cancel all applications
  - form RS
  - do bus reconfiguration
  - drop any non-target GPCs
  - prepare for overlays (ARC_MEM_OVL)
    -- select bus to use
  - issue SVC call (FCOS) to transfer 1st phase
  - if errors, switch to alternate bus, reissue SVC call
  - issue SVC call (FCOS) to transfer 2nd phase if required
  - if need MFB as well as OPS, now call ARC_MEM_OVL again to get OPS overlay
  - if errors, switch to alternate bus, reissue SVC call
  - reset to GPC/MMU if MMU/MMU had been selected
  - clear CAM
- Initiate OPS

2. New OPS request
   - Verify valid keystrokes
   - Verify no transition in progress
   - Verify validity of request using (DM6V_TR_TAB)
     - if not valid check combination transition table (DM6V_COMBINATION)
   - Check applications moding table (AMT) to see if required OPS is in core (i.e., OPS 1-6)
   - Check GPC reconfiguration table (GRT) to see if mode is legal for MF and OPS requested

- Determine target GPCs and run GPC sets (DM6_OPS)
  - check table set in GPC memory and NBAT target entries
  - must be in CS, in OPS 0, and targeted
- Process pre-position (ARC_GPC)
  - determine if there are any source GPCs
  - select 2 sources if need OPS and MFB overlays
  - pick lowest ID GPC in current set if source exists (primary GPC), otherwise use lowest ID GPC of secondary GPCs
  - if G9-G1 and Archive load set, position for G3 vice G1
  - if MMU/MMU selected, ignore GPC sources
  - if GPC/LDB selected, and no GPC source, then set error condition
  - if no GPC source exists, or MMU/MMU selected, or G3 archive retrieve not selected, then request pre-positioning of both MMUs
    - -- use table of phase # vs. MC # (CZ2V_GRT_MC_PHASES)
    - -- use in-core phase table to find MMU address
    - -- issue SVC calls to pre-position both MMUs
- Recognize completion of pre-positioning (DM2_APP)
  - requires at least one MMU to not have any errors against the transaction
- Take set to OPS 0 (DM2_REC_PREP)
- Schedule new OPS (DM5_NEW_CSEG)
- Initiate overlay process (ARC_GPC)
  - determine source GPCs and build overlay masks
  - cancel all applications
  - form new RS

- do bus reconfiguration
- drop any non-target set GPCs
- prepare for overlays (ARC_MEM_OVL)
  -- select bus to use
- issue SVC call (FCOS) for 1st phase required *
  -- if error, switch to alternate bus, reissue SVC
- issue SVC call (FCOS) for 2nd phase, if required
- if need MFB also, call ARC_MEM_OVL again to get OPS overlay
- reset to GPC/MMU if was on MMU/MMU
- clear CAM

• Initiate new OPS

3. FD request

• Verify valid keystrokes

• Verify valid entries for GPC MEMORY DISP

• Start FD item processing (ARG_GPC_CONFIG)

-verify:

-- GPC not in a RS

-- GPC is in OPS 0

-- GPC not in a reconfiguration status

---

* If G3 retrieve case, special SVC call issued

- if all targets have G3 archive, then all transfer to low core
- if not all targets have G3 archive, those with G3 transfer it down to low core and low ID GPC transfers it to GPCs without archive

If G9-G1 and G3 archive load requested, load G3 overlays, shift to upper core, set archive present flags if no errors, clear any errors, then continue with standard G1 load

- Start FD overlay process (ARC_GPC_RECONFIG)
  - force target set to this GPC only
  - set flag to indicate required OPS not resident
  - if need MFB, go to MMU (ARC_MEM_OVL)
    -- select bus to use
    -- issue SVC call (FCOS) for MFB overlay
    -- if error, switch to alternate bus, reissue SVC
  - call ARC_MEM_OVL to get OPS overlay
    --select bus to use
    --issue SVC call for 1st phase of OPS
    --if error, switch to alternate bus and reissue SVC
    --issue SVC call for 2nd phase of OPS, if required
- Initiate OPS 0

B. Flow for loading From MMU

The flowcharts of figure 11.3.2-3 represent a functional overview of an OPS transition when MMUs are the source of the overlays. (These charts were developed by Steve Hamm of the Training Division. They are intended to represent the flow as of OI-20 software.) Note that the flow is only valid for an OPS transition and not for a FD load or any other MMU use.

11.3.2.5 DMC SUPER

TBS

11.3.2.6 FCOS Modules

Once the applications programs decide that overlays need to be transferred from MMUs or from other GPCs, then SVC calls are made which bring the FCOS modules into play. These modules handle all the setup prior to actual software transfer, the actual transfer of software, and the error checking before and after the transfer of software. The overall process flow of FCOS modules is shown in figure 11.3.2-4.

This shows only the main flow and does not show any error handling programs that are also part of the operation. The discussion will start at the top (calling level) and progress down to the actual MMU BCE programs.

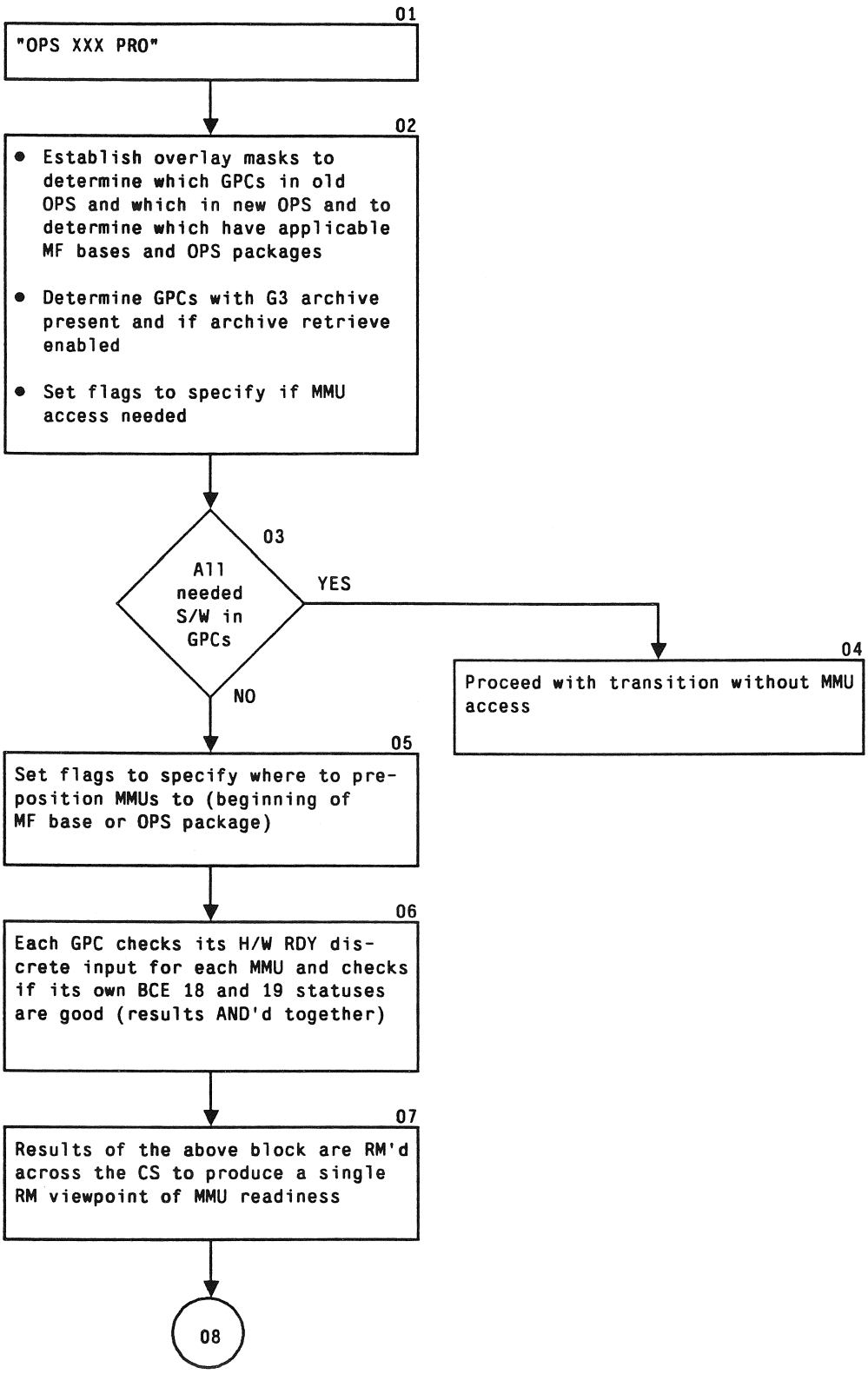


Figure 11.3.2-3.- OPS transition from MMU.

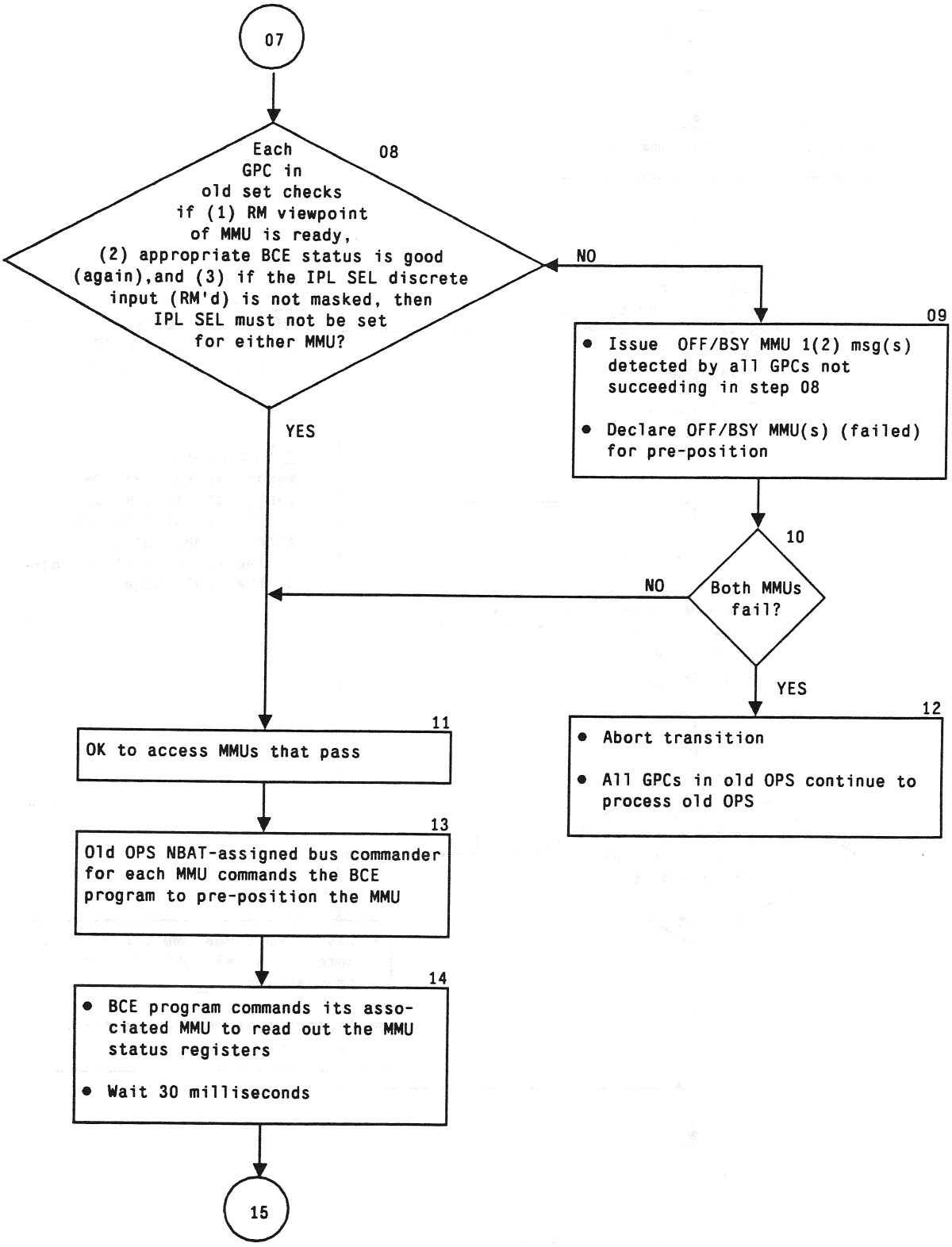


Figure 11.3.2-3.- Continued.

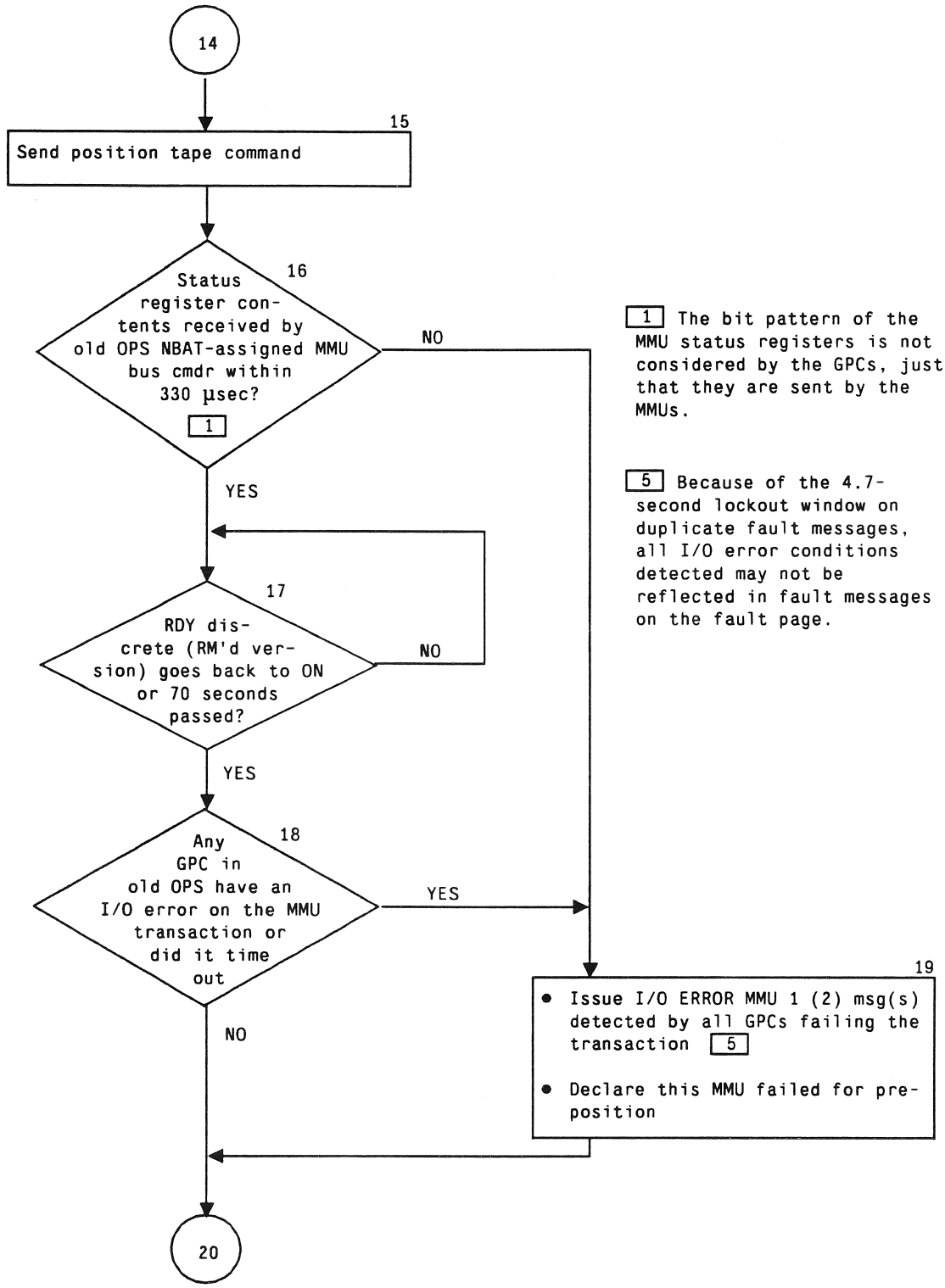


Figure 11.3.2-3.- Continued.

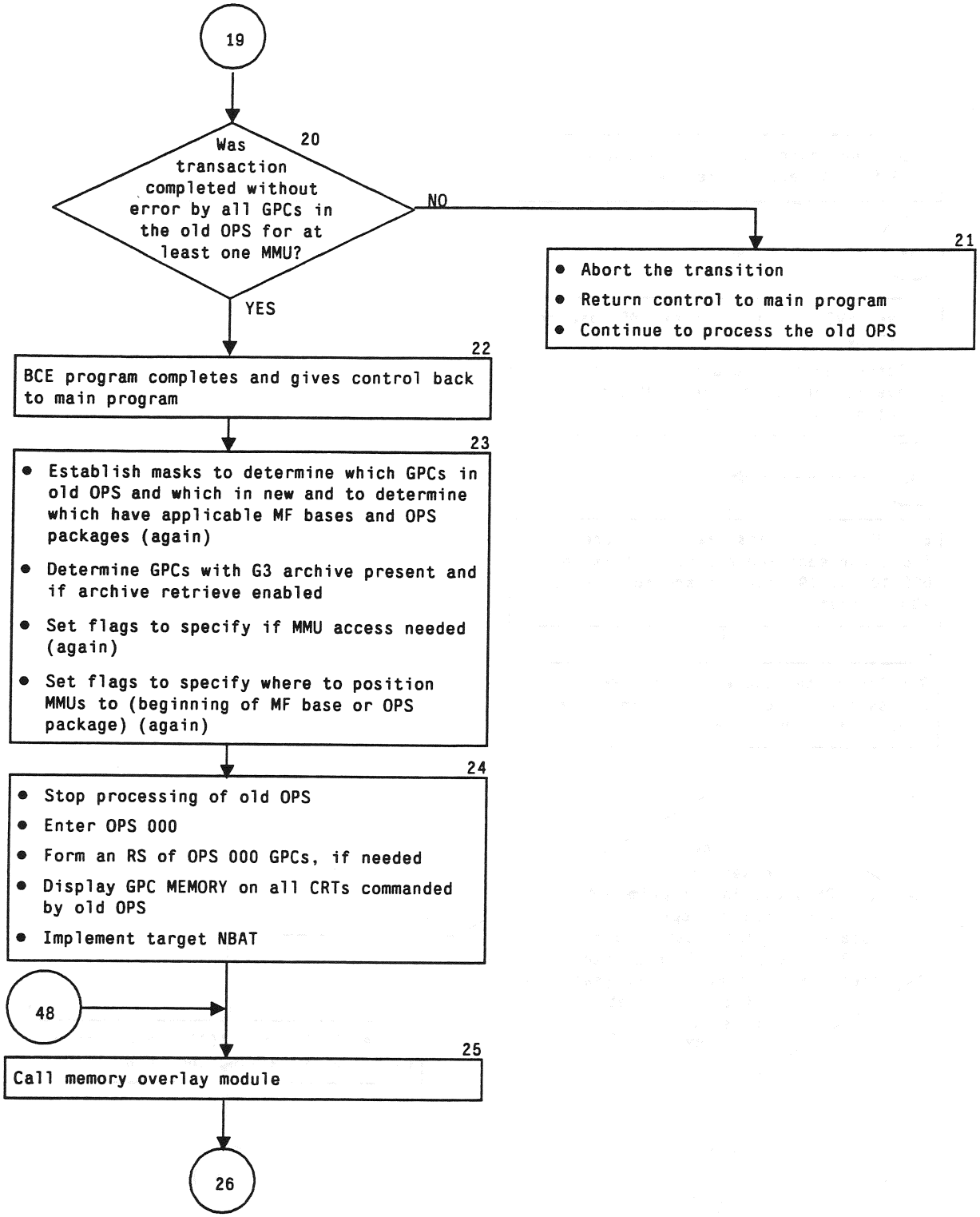


Figure 11.3.2-3.- Continued.

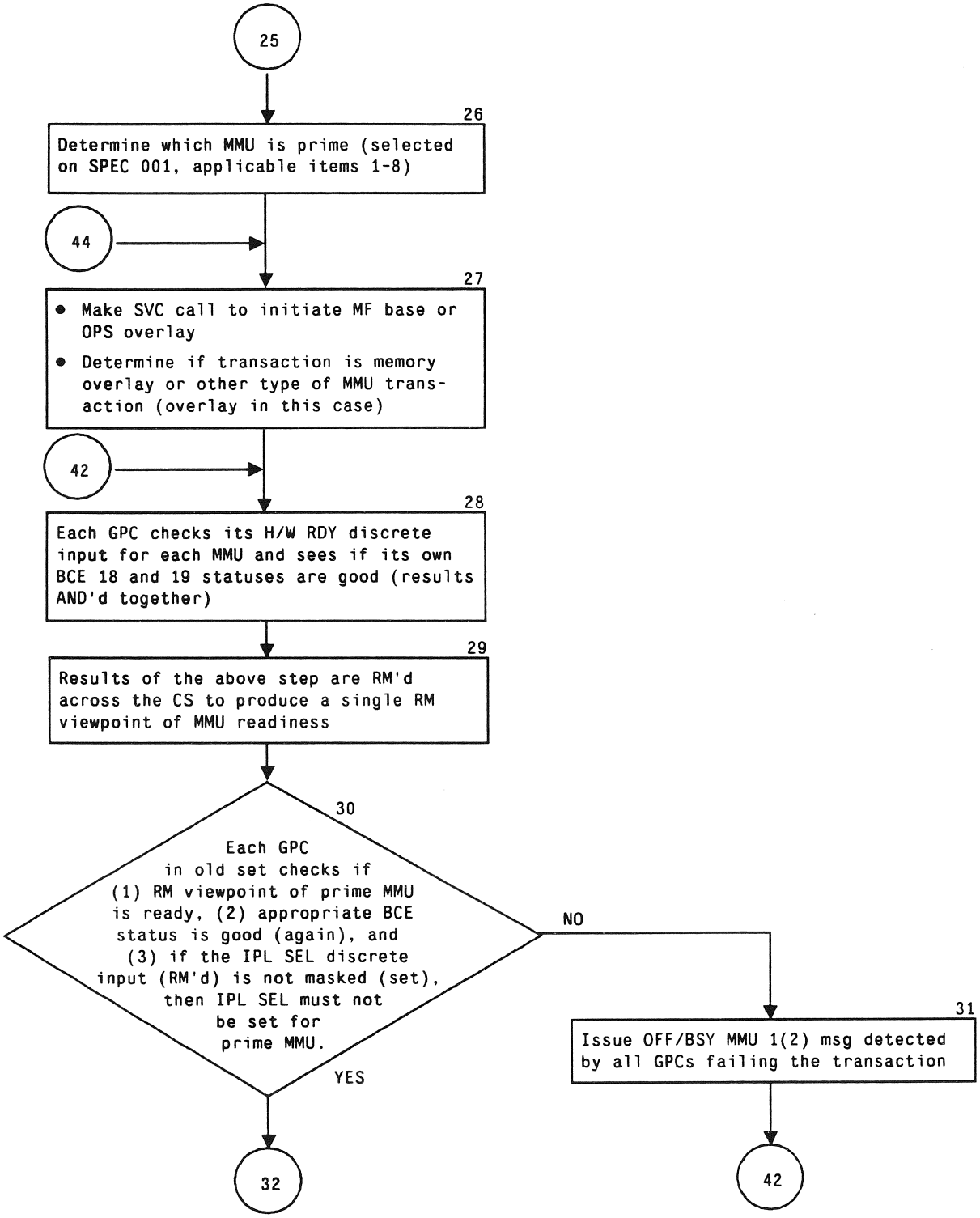


Figure 11.3.2-3.- Continued.

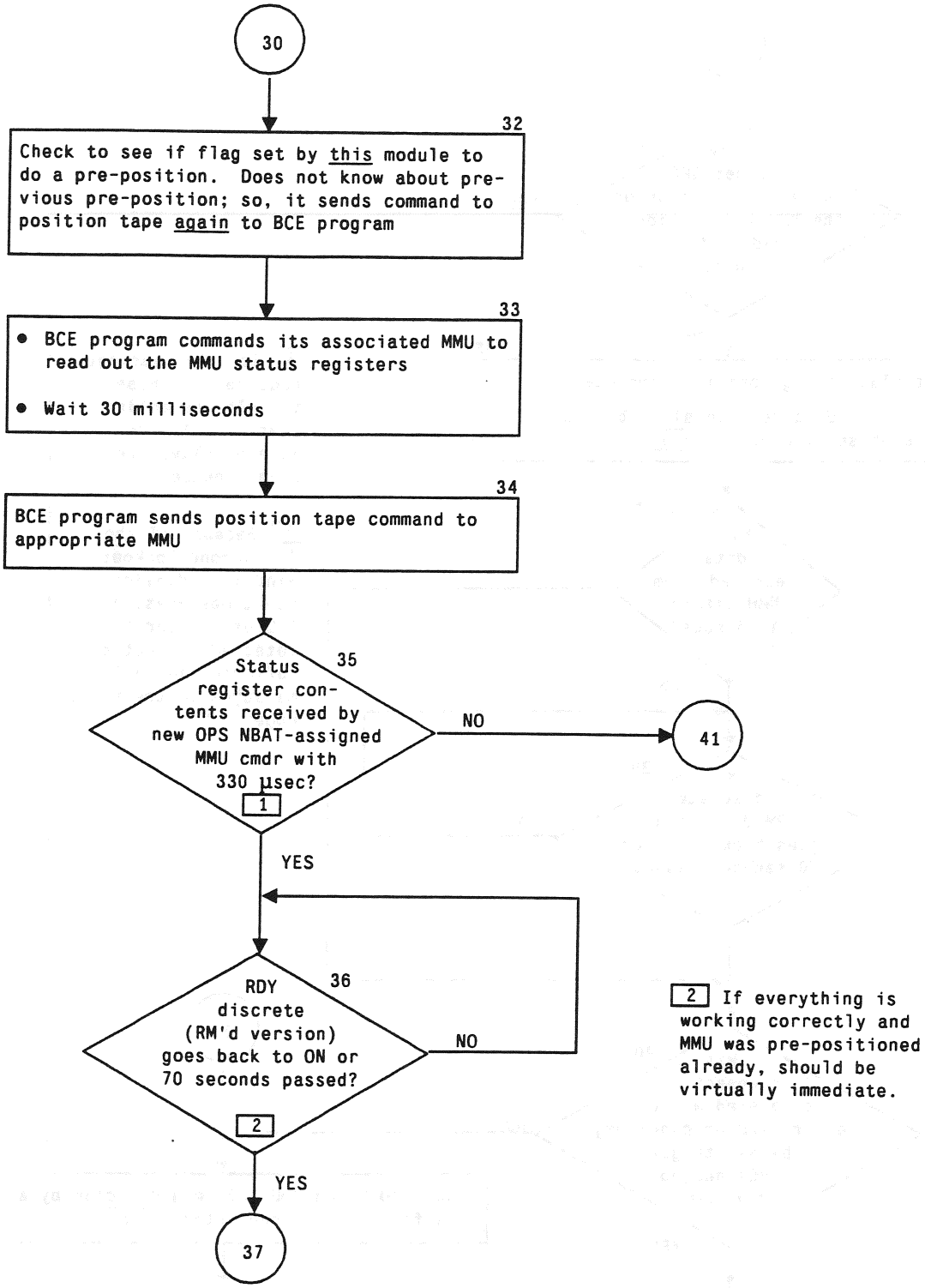


Figure 11.3.2-3.- Continued.

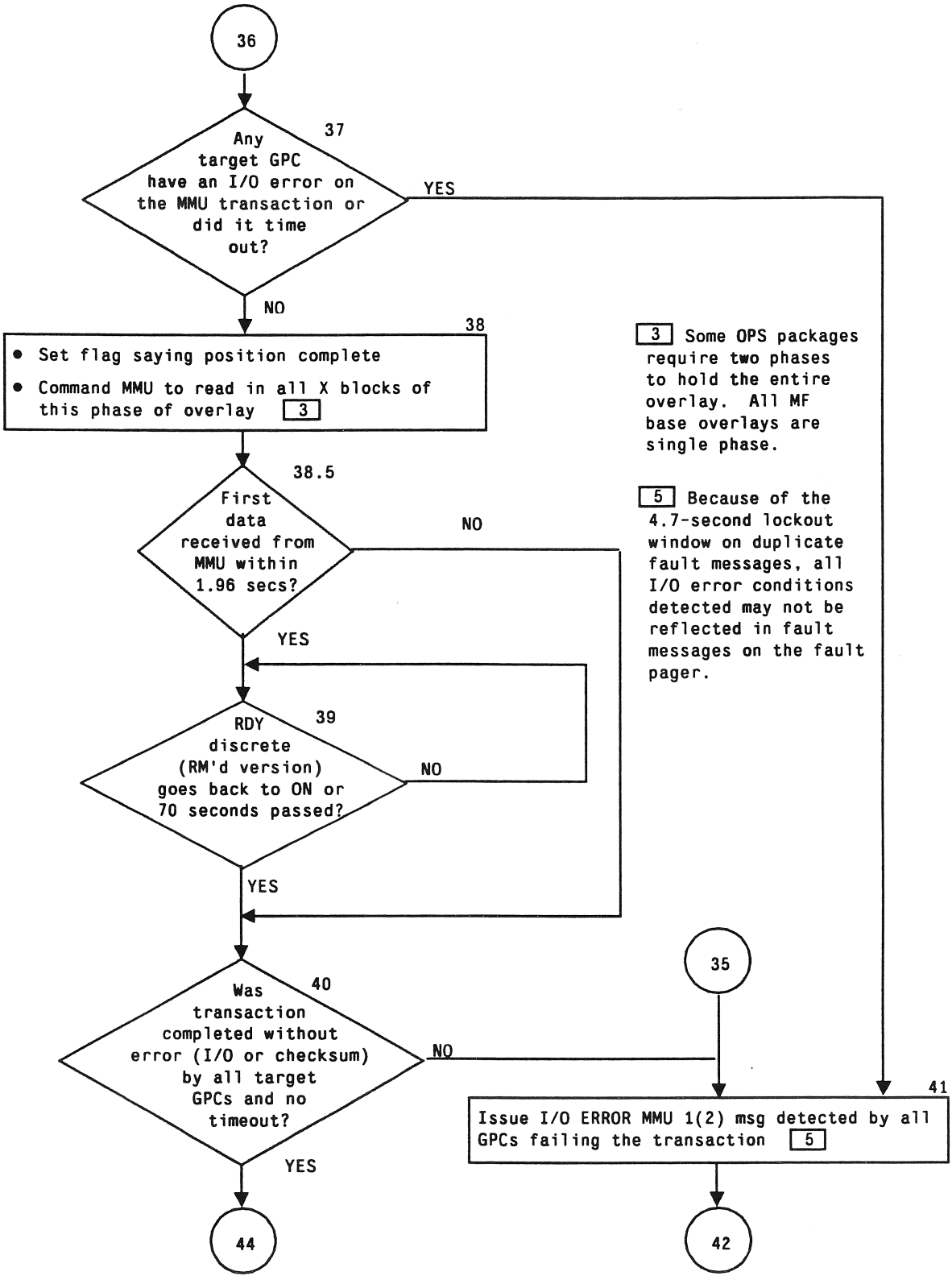


Figure 11.3.2-3.- Continued.

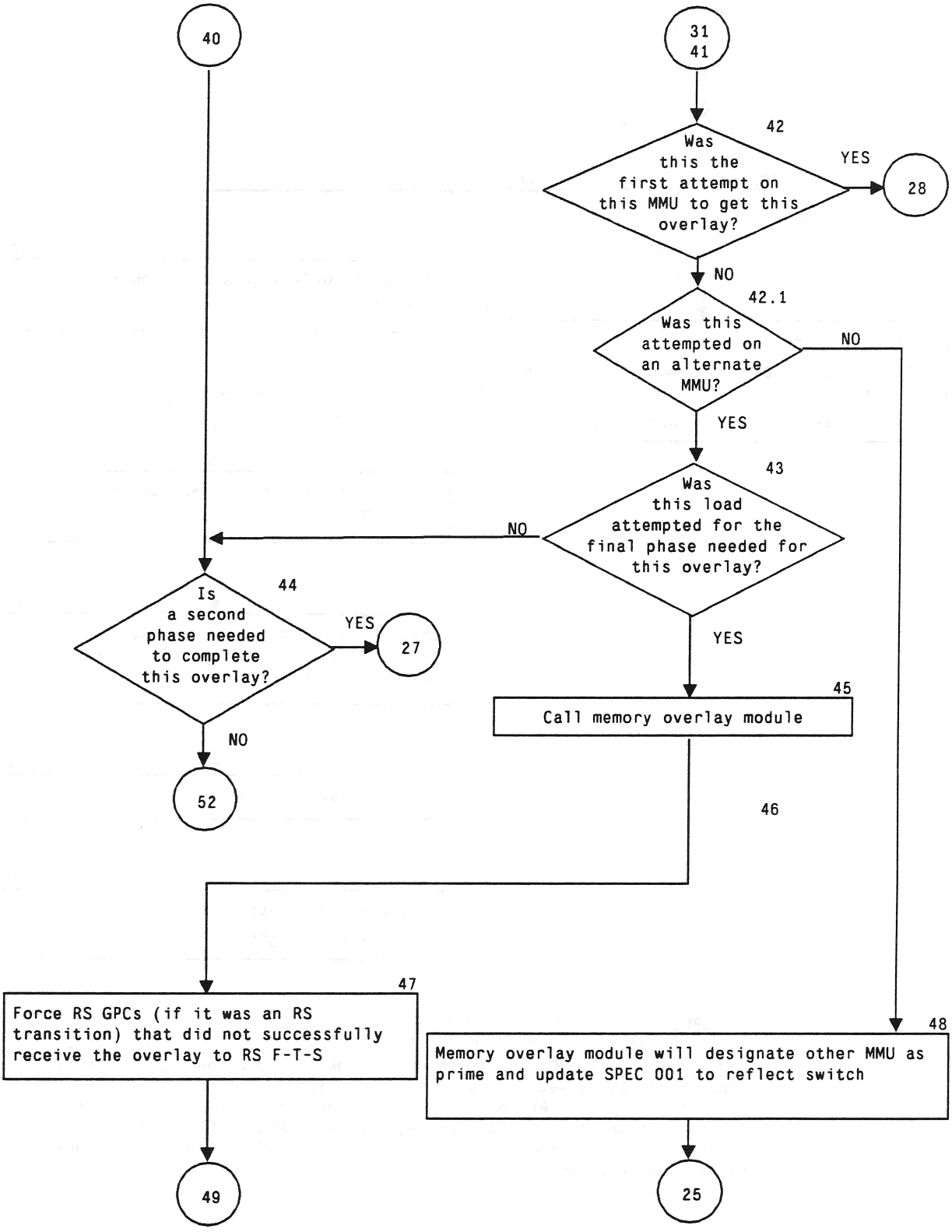
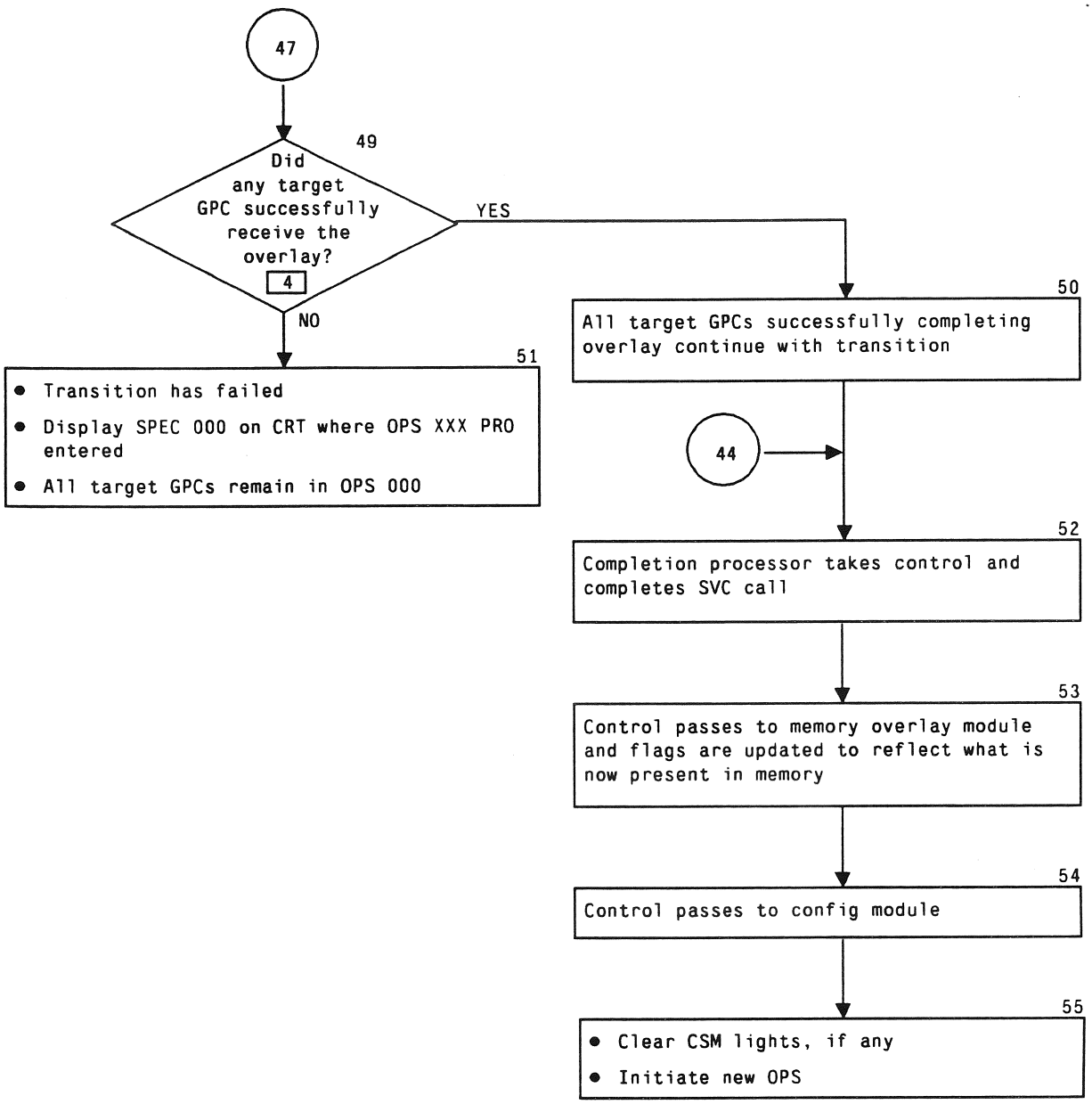


Figure 11.3.2-3.- Continued.



4 GPCs 1 and 2 are processing OPS 201 GNC. GPC3 is a G2FD GPC. It has been brought to RUN in OPS 000. The OPS 3 NBAT is set up for GPCs 1, 2, and 3, and an OPS 301 PRO is entered. Unknown to anyone, GPC 3 has both BCEs 18 and 19 failed. GPCs 1 and 2 successfully pre-position the MMUs and enter OPS 000. At this point, GPC 3 has errors on a try and retry for both phases of the OPS 3 package for both MMUs. GPC 3 is forced to RS F-T-S and is dropped out of the transition. GPCs 1 and 2 continue the transition successfully.

Figure 11.3.2-3.- Concluded.

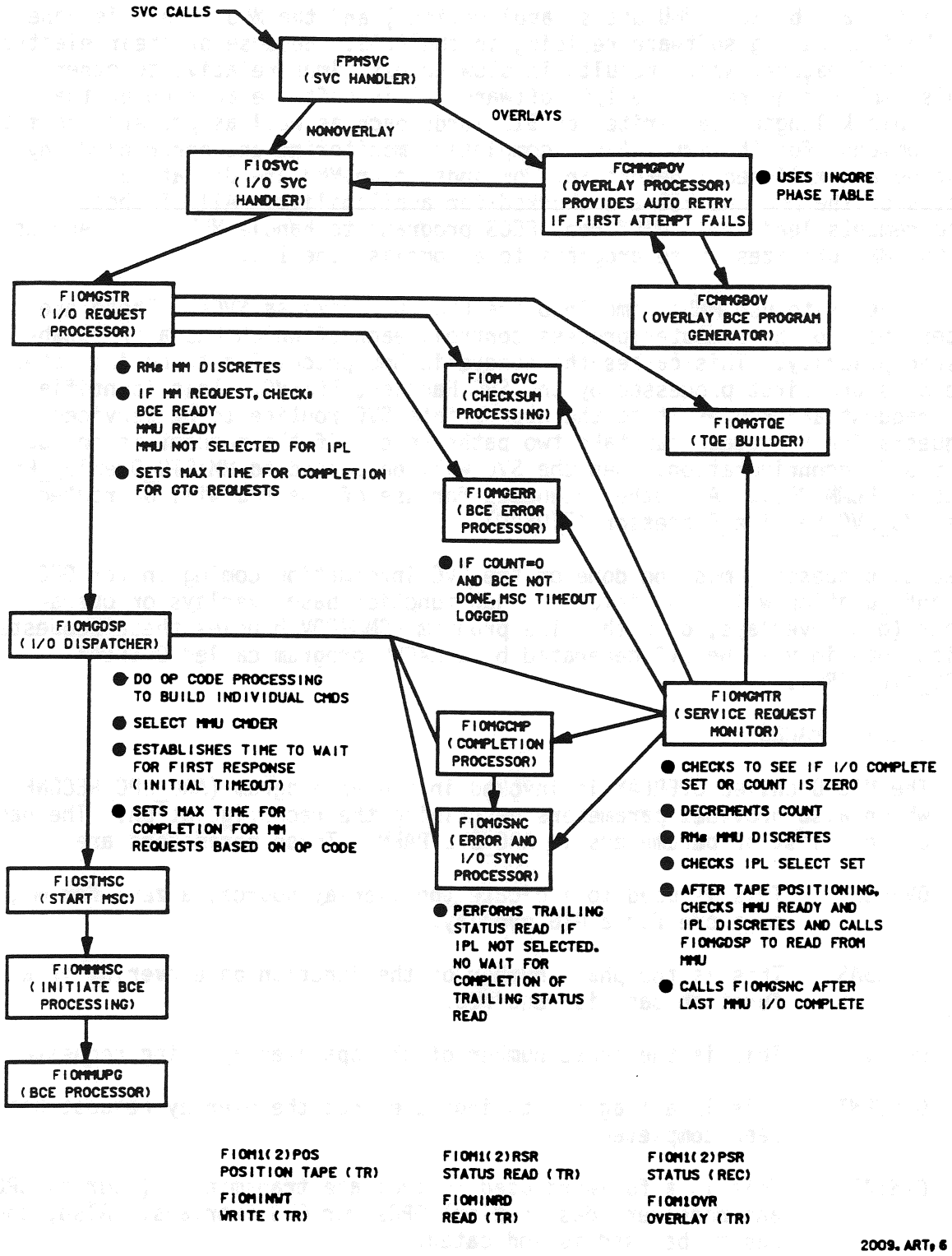


Figure 11.3.2-4.- FCOS software flows for transitions (only main paths are shown).

The interface between MMU users (applications) and the MMU itself is done by the interfacing software residing in the FCOS. Because of their electromechanical nature, which results in slow access times relative to other BTU's, MMU's require unique I/O software. This software must cover the fixed block length read/writes of 512 words each as well as properly format the commands for the MMU. Also, completion monitoring and error handling must be accomplished. Before any commands to an MMU are dispatched, the status of the MMU and BCE are checked for availability. All of these requirements lead to 13 different FCOS programs to handle MMU I/O. Any user of the MMU utilizes these programs to accomplish the I/O.

All requests to use MMUs come into the FCOS programs as SVCs. These are interrupts to the computer process control, each of which has a preestablished priority. This causes the noncyclic MMU processing to be initiated. The SVCs are first processed by the SVC Handler, FPMSVC. This identifies the request and routes it to the appropriate SVC routine to be serviced. Requests for MMU usage can take two paths here. If the request is to support GPC reconfiguration, then the SVC will be passed to MM/GTG_Overlay_Processor (FCMMGPOV). All other requests for use of the MMU will be routed to the I/O_SVC_Service_Processor (FIOSVC).

Special processing must be done on the SVC information coming in for GPC reconfiguration which may involve major function base overlays or operations (ops) overlays, or both. The program FCMMGPOV handles these requests, which come in via the SVC generated by a MACRO program called OVERLAY (ARC_OVL_PARM).

A. Overlay MACRO

The MACRO called OVERLAY is invoked in a user program (ARC_GPC_RECONFIG) which also provides parameters describing the reconfiguration. The name of this list of parameters is ARC_OVL PARM. These parameters are

OVSRCE — This is used to indicate the overlay source; a zero for MM, or a one for a GTG overlay.

OVFNBAS — This is the phase number of the function base overlay if a function base is required.

OPSOV — This is the phase number of the ops overlay being requested.

OVEVENT — This is a flag set to indicate that the overlay request has been completed.

OVSTAT — This is a fullword used to indicate transmitter (source) GPC and receiver (destination) GPCs for GTG overlays. Also, the bus to be used is indicated.

	Source GPC					Destination GPC							LDB 1	LDB 2		MMU bus		unused
Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13		18	19	20-31
GPC	1	2	3	4	5	1	2	3	4	5								

Because this MACRO provides only phase information and the MMU knows only file, track, subfile, and block, additional processing is required. This is done by FCMMGPOV using the in-core phase table (#PFCMGPT).

B. Overlay operations

The overlay processor, FCMMGPOV, calls another program, FCMMGBPV (MM/GTG overlay BCE program generator), in order to provide the real-time BCE receiver programs needed during phase overlays. This program also generates the sequences to handle the overlaying of the data into main memory, as well as the unprotecting and reprotecting of memory. Control is returned to FCMMGPOV.

In the overlay process, it is FCMMGPOV which causes the automatic retry of an overlay transaction if the first attempt fails. This is valid whether the overlay is a GTG type or comes from MMU and is done whether LDBs or MMU buses are being used.

The next step in the chain of programs is a call to the I/O_SVC_Service_Processor (FIOSVC) by either FCMMGPOV if an overlay, or FPMSVC if other MMU use. FIOSVC initializes the I/O operation and passes the request on to the MM/GTG_I/O_Request_Processor (FIOMGSTR).

C. Discrete Redundancy Management

The request processor (FIOMGSTR) performs more initialization and then does some preliminary checks before dispatching the I/O request. The first thing it does is call the GPC Discrete_Redundancy_Manager (FCMDSCRM) to process the MM IPL select discretes and MM ready discretes contained in CZ2B_DIA1. This involves checking to see if the majority of common set GPCs (>1/2) agree on the status of these discretes. Only if the majority agree that a discrete has changed from the last agreed-to state is the change entered into the redundancy managed discrete word, CZ2B_DIA_RM1.

Next, the I/O request is checked to see if this is a GTG transaction or an MMU request. If it is a GTG request, then the overlay buffer needed is checked to see if it is available and if not, MMU OFF/BUSY is annunciated. For MMU use, the BCE to be used is first checked. If it is busy, MMU OFF/BUSY is annunciated. Next, the MMU ready bit in the redundancy managed word is checked. If it is not ready, MMU OFF/BUSY is annunciated. Finally the MMU selected for IPL bit is checked in the redundancy managed word. If it is selected for the required MMU, then MMU OFF/BUSY is annunciated. If any of these conditions occurs, the processing is terminated with the annunciation of the fault message.

Another part of the processing needs to be explained but does not really have any effect. The MM/GTG_BCE_ERROR_Processor (FIOMGERR) is called to check the BCEs status to see if any problem occurred the last time that the BCE was used. This would have been for an MMU status read, which is always done at the end of MMU transactions. This trailing status read has no error processing or logging associated with it at the time of occurrence. When the BCE is checked during the start of the next I/O, error processing is done, but no error logging is done here either. The net effect resulting from the error processing is a resetting of the BCE to ready it for the new I/O operation.

D. Timeout calculations

If nothing has gone wrong so far, then FIOMGSTR is ready to dispatch the request. Before dispatching the request, a timer count is generated in order to monitor the execution and know the maximum time to wait for the operation to finish. For all MMU operations, except BSR read and position tape, the count is set to 50. This is 50 times 200 ms, or a total possible wait time of 10 sec. For the BSR read case, the count is set to 2 (400 ms), and in the position tape case to 350 (70 sec). If a GTG transfer is requested, then the count is calculated based on the number of loadblocks and total number of words.

For the GTG transfer case, the BCE program execution time is based on 16.5 μs blocks and the number of time blocks is calculated in FCMMGBOV. Each loadblock is considered separately and is subdivided into 2048 word groups, if necessary. The count generated for each loadblock is based on the following expression, where N is the total number of halfwords in a given loadblock, INT is the integer portion of the resulting quotient, and TB is the number of time blocks for this loadblock.

$$TB = 23\,(INT\left(\frac{N}{2048}\right) + 1) + 2N$$

The number of counts is calculated as shown below:

$$COUNT = \left(\frac{TB \times 16.5}{200{,}000} + 1\right)$$

For a typical ops transition involving an ops overlay, a count of 20 is calculated which equates to 4 sec.

After these timeout limits have been calculated, the MM/GTG_I/O_Dispatcher (FIOMGDSP) is called to do further processing of the I/O request and partial building of the MMU command words before the Master Sequence Controller (MSC) is allowed to start the process.

E. Command processing

The first thing done by FIOMGDSP is to halt and reenable the BCE to be used for this transaction. This ensures that the BCE registers are clear and that the BCE is reset and ready for use. After this FIOMGDSP splits into GTG requests and MMU requests. This involves building the Program Control Output (PCO) instruction word which will cause the BCE to execute the proper set of instructions.

If the MMU is to be used, additional steps are executed. A tape positioned flag is checked to see if the tape has already been positioned. If not, then an op code of 8 is set to cause positioning to be done. If the tape has already been positioned and the request is for MMU read or write only (not overlay, status read, or positioning), then processing is done to get the address of the I/O buffer in the right format and stored in the BCE register table. Next, MMU op code dependent processing is done for each type of MMU command. This process consists of taking the skeleton command word for each case and OR'ing in the address data of file, track, subfile, block, and number of blocks. This type of information is supplied by either the individual applications programs, or if an overlay, from the FCMMGPOV program which has already been discussed. Now on to the individual command processing.

1. Write CMD processing

The write processing (FIOMMNWT) starts with the MMU skeleton command word for write enable, which contains the op code "1010." To this is added the track information. The write command skeleton word has an op code "1000" and a dummy block count of zero. (How or where this gets changed to a nonzero value is not clear.) The track, subfile, and block are OR'd into this skeleton word and then it is stored into a BCE command table.

2. Utility CMD write processing

The other write processing module is called MM_Utility_Write_Request Processor (FIOMMMUW). It is set up to handle writes of up to 32 blocks of data and is used for capability 1 type writes only (ground commands up the LDBs). This program builds a write enable command word by adding the track address to a write enable skeleton command word. The word count for the transaction is decremented by 1 and then divided by 512 to get a block count to use in the extended block count command word. Next, multiple blocks to be written are handled by generating and storing addresses for each MMU block into the Mass_Memory_Utility_Base_ Register. The write command word is built by adding the track subfile and block information to the skeleton word. The op code is changed to cause a read to verify the write (code 3).

3. Overlay CMD processing

The processing for op code 7 is done in the MM Overlay Read Request Processor (FIOMMOVL). It builds the extended block count command by using the block count found in TIOQWDCD. Normally this would contain a word count, but for overlay processing, the FCMMGPOV program calculates the block count and places it in this word. Next the read command is built by OR'ing in the track, subfile, and block information found in the variable TIOQSTAD. This comes from the FIOSVC program where the information is transferred from TIOSSTAD which is set in FCMMGPOV where it was read from the in-core phase table. Finally, the flag indicating that the last MMU transaction has been completed is set to one.

4. Position tape CMD processing

Op code 8 contains the position tape command processing. The command is built by starting with a skeleton word containing the MMU op code for positioning the tape (000). The track, file, and subfile information is obtained from the MM address stored in TIOQSTAD. The track and file are directly OR'd into the command word. The subfile, if nonzero, is decremented by one before being put into the command word. If subfile zero is specified, then the mask for the BOF bit is used instead of the subfile information. This subfile adjustment is done in order to position the tape to the subfile (of BOF) area preceding the subfile to be read from or written to.

F. MMU bus/commander selection

After the individual op code processing has been completed, FIOMGDSP does MMU commander selection. Although the process is called commander selection, it really sets up a GPC as a listener if it is not assigned to be a commander. If there is no redundant set in existence, then no commander selection is done here. Before commander selection is done, the bus (which dictates the MMU) to be used must be known. The way this information (bus mask word) becomes available to this module is far from clear and therefore warrants an explanation.

There are two ways for the bus masks to be built depending on whether the I/O is an overlay transaction or nonoverlay use of the MMU. For the overlay case, the mask word starts in an applications program called ARC_GPC RECONFIG (GPC reconfiguration). This program sets up the masks and parameters used during a reconfiguration of the GPCs. One of these contains bits to designate the MMU bus to use (if required for the overlay). The word is one of the overlay MACRO parameters ARC_OVL PARM. OVSTAT is referenced in GPC reconfiguration as ARC_OVL REQ. (The layout of this parameter was discussed in the writeup of the module FCMMGPOV.) The FCOS can then access this parameter since it is passed across from applications to FCOS as part of the SVC call. The FCMMGPOV program picks up this mask and calls it TOPLSTAT. It then establishes another name, TIOSBUFA. The FIOSVC program picks up the parameter TIOSBUFA and

calls it TIOQBUFA. Finally the MM/GTG_T/O Request_Processor (FIOMGSTR) picks up TIOQBUFA and puts it into the MM BCE mask word (TIOQMNTM).

If the I/O is a nonoverlay use of the MMU, then FIOMGSTR goes to the MMU major function (MF) table (CZ2B_MM_MF) to obtain the MMU assigned to the MF doing the I/O. The table contains one halfword for each possible MF (PL, GNC, SM, or ops 0) with bit 3 set if MMU 1 is selected and bit 4 set if MMU 2 is selected. This table is set by item entries (item 1-8) to the DPS UTILITY display or by application program manipulations. The MMU obtained from the appropriate word out of this table is then loaded in TIOMGSTR.

Now that the MM BCE mask word TIOQMNTM has been established, we can press on with the selection of the MMU commander performed by GIOMGDSP. The NBAT is used to determine the commander for this MC for the MMU specified in TIOQMNTM. (The NBAT contains a list of GPC commanders for flight critical strings, CRTs, MMUs, LDBs, and payload buses for each memory configuration.) If the GPC from the NBAT entry is the self GPC (i.e., GPC doing this processing), then self GPC is set as the commander of the bus and no further selection occurs. If the NBAT-specified commander is not the self GPC, then the op code for the transaction is checked. If an overlay or position tape activity is specified by the op code, then default commander selection is done. For this process, if the NBAT specified commander is not in the RS, then the lowest ID GPC of the RS (including self GPC) will be selected as the commander and other GPCs will be listeners.

The default commander selection process was first added to the software in release 18. It was originally intended to be a default commander selection for all uses of MMU, not just overlay and position tape transactions, with the purpose of preventing failing an OPS transition because the NBAT-specified GPC had failed out of the set. One problem in having default commander on the overlay read and position type transactions only was with the trailing status read command done as the last command in each MMU transaction. With no default commander selection, if the commander specified by NBAT was not available, the BCE would wait its entire timeout period of 1.96 seconds before timing out, waiting for a response from the MMU. This response never comes because no GPC ever commanded the status read. Because this trailing status read is not monitored for completion, after the command to read status is processed (it thinks it has been started), the software presses on with the next operation. The next operation finds the BCE busy (still waiting) and cause that operation to fail. The proper fix is to make the default commander selection apply to all uses of the MMU. This is planned for release 20 software. The release 19 software has a change in FIOMGDSP that provides an interim fix. The BCE timeout for only the status reads was cut to 330 μs. This is an adequate time for the MMU to return the status words and does not tie up the BCE longer than necessary if the MMU has died or the commander for the read status operation does not exist.

Having been buried so deep in FIOMGDSP for several paragraphs, it is now time to step back and get out of it. The main things accomplished were the processing of the individual op codes and the MMU commander selection. After these things are done, the final activity of FIOMGDSP is to do a call to start the MSC processor.

G. Completion monitoring

One of the sideline programs which is still important is the MM/GTG IO_Service_Request_Monitor (FIOMGMTR). This program monitors outstanding MMU I/O transactions or GTG transactions to determine if the operation has finished and, if so, to do completion processing.

If the transaction is a GTG, then the monitoring is fairly straightforward. If the monitor count (set in FIOMGSTR) is not zero, it is decremented one and checked to see if it is now zero (which means that the operation should be complete and a flag is set indicating I/O complete). Checksum verification is done to check on the validity of the data transferred, and if an error occurred, a checksum error is set. The BCE error processor (FIOMGERR) is called to check on any BCE NOGO or MSC timeout that may have occurred. If the I/O complete flag is on because the count reached zero or the transaction finished, then the flag is reset and the completion processor (FIOMGCMP) is called.

For MMU usage, much more processing is required. First the MMU discretes are redundancy managed by calling FCMDSCRM. Then the MMU ready discrete is checked to see if the MMU is still busy. If it is busy, the IPL select discretes are checked to see if this MMU is selected for supporting on IPL. If it is selected for IPL, then the I/O completion flag is set and an I/O error logged against the transaction attempt, and completion processing is done again.

If the MMU is busy but not selected for IPL, then the wait count is decremented one and checked to see if it has reached zero, indicating that the operation should be finished. The error processor (FIOMGERR) is called to check for BCE problems and a fail/powered down error will be logged and the completion processor called.

If the MMU is found to be ready, then the idle processing is done. If no more I/O transactions are required, the checksum processor FIOMGCV is called to generate and verify the checksum on the transferred data. If a problem occurred, a checksum error will be logged. The BCE error processor (FIOMGERR) is called to check for any BCE problems, and then normal completion processing (FIOMGCMP) is done.

If the MMU is ready and more I/O transactions are required, the IPL select discrete is first checked before allowing another transaction to be dispatched. If the IPL select bit is set, then the I/O completion flag is set to force the completion processing to be run. If the IPL select is not set, then FIOMGERR is run to check for BCE errors and FIOMGSNC is run to do error and/or sync processing before dispatching the next I/O transaction.

H. BCE programs

Finally, in the main line of programs accomplishing MMU I/O is the Mass_Memory_BCE_Processor (FIOMMUPG). This csect is a collection of the BCE programs needed to read, write, position tape, or read status from an MMU. These BCE programs are not called, but rather they are invoked by having their addresses stored in the MMU BCE program counter by the I/O dispatcher (FIOMGDSP), and then the BCE is started by the MSC processor (FIOMMMSC).

1. Position tape

   The position tape transmitter program reads the status words of an MMU and then sends the prebuilt position tape command.

2. Read status

   The read status transmitter program sends out a request for the status words and then stores them.

3. Status word receiver

   The status word receiver program is used by listening GPCs to hear the status words read by either a position tape command or a read status command.

4. Write

   The write BCE program issues a write enable command followed by a write command. It receives the status complete word (SCW) from the MMU and stores it. It then transmits one MM block of data (512 halfwords) to the MMU and stores the BCE status before returning to a wait state.

5. Read

   The read program contains a transmitter section and a receiver section. The transmitter section issues an extended block count followed by a read command. It then branches to the common part of read receiver program. The read receiver program contains delays to get it in step with the transmitter program. After the delays, the common part of the receive program receives the data from the MMU and then stores the BCE status word before returning to a wait state. Note that this read program is for all reads other than overlays.

6. Overlay read

The overlay read program contains sequences for the transmitter and receiver modes. The transmitter section issues an extended block count followed by a read command. The program then branches to an address in the MMU BCE branch table to execute the dynamically created receive sequences, which are built in the overlay BCE program generators (FCMMGBOV). When the overlay is read into memory, a branch back to the overlay BCE program occurs and the BCE status is read and stored. The overlay read receiver program provides a delay to get it in step with transmitter programs and then branches to the same BCE address tables as the transmitter sequence does.

7. Write capability 1

There is another MMU write BCE processor called FIOMUWPG, which will not be covered. This program is used only for capability 1 writes to the MMU. These are writes to the MMU over the LDBs done on the ground only. These address the MMU by FTSB instead of by phase and loadblocks.

I. Error/completion processing

After an I/O operation completes or times out, completion processing is done by MM/GTG_I/O_Completion_Processor (FIOMGCMP) by calling it from the service request monitor (FIOMGMTR). This may also be called by the I/O request processor (FIOMSTR) if an I/O request cannot be started. All this process does is call error processing (FIOMGSNC) and release resources for other uses in addition to doing cleanup and resetting.

There are a few other FCOS MMU programs that are not mainline programs but are still important to the operation. One of these programs is the MM/GTG_ MSC_Processor (FIOMMMSC). This program is an MSC routine that initiates MMU BCE processing. This routine is called by the start MSC_Processor (FIOSTMSC) and exits to the monitor routine (FIOMNTR).

Another of these sideline programs is the MM/GTG_BCF_Error_Processor (FIOMGERR). This program provides BCE and MSC error detection processing. It may be called by the service request monitor (FIOMGMTR) when an I/O transaction completes or the interrupt count expires. It is also called by the I/O request processor (FIOMGSTR) to check the BCE status from the last status read operation.

The other error handling processor is MM/GTG_Error_And/or_Sync_Processor (FIOMGSNC). This program handles processing if errors occur during I/O and ensures that GPCs stay together during the I/O by calling the I/O sync processor (FCMISYNC). If an I/O transaction completes and more transactions are needed to satisfy an I/O request, then the service request monitor (FIOMGMTR) program calls FIOMGSNC. If the last transaction needed to satisfy the I/O has been completed, then the completion processor (FIOMGCMP) calls FIOMGSVC. If errors have been detected, the level C error handler (FIOERRLC) is called to log and annunciate the

error. For overlay transactions, if an error occurs, the identity of the GPC is stored in the Transactions Status Word (TSW) along with the Input Program Report (IPR) status. If the transaction is not an overlay, then a transaction failed indicator is set in the TSW. If the transaction is the last to satisfy an I/O request, then FIOMGSNC calls the dispatcher (FIOMGDSP) to initiate an MMU bite status read.

At the end of MMU or GTG I/O operations, a program called MM/GTG_Checksum Processor (FIOMGCV) is called to do checksum generation for each loadblock. This is done by masking out any overflow bits and summing the halfwords. This sum is then compared with the checksum stored in the last halfword of the loadblock. This process is called by the I/O service request monitor (FIOMGMTR).

11.3.3 GPC Initialization - Introduction

GPC Initialization is defined as the orderly sequencing of a GPC from an inactive state to an operational state. Of course, there must be sufficient hardware and software checks to assure the user of an "acceptable" confidence level regarding the success of the operation. GPC initialization includes functions such as loading system software in the GPC from mass memory, initializing the hardware and software, initializing the Input/Output Processor (IOP), and verification of the GPC's health. There are two basic types of GPC initializations supported by the Flight Computer Operating System (FCOS).

The first type is an Initial Program Load (IPL) initialization that occurs when a user supplies power to a GPC in HALT mode, performs the IPL, then places the GPC in the STBY mode. This option requires hardware/firmware support, as well as system software support. The GPC MODE (HALT, STANDBY, and RUN) and IPL hardware switches are used to control the IPL sequence.

The second type is a Normal Initialization, performed when a previously loaded (IPL'ed) GPC is swithed from HALT to STBY with the power ON or is powered up with the mode switch in STBY.

This non-IPL initialization is supported totally by the system software.

11.3.3.1 IPL Initialization

The IPL initialization of a GPC is initiated by hardware/firmware in response to switch positions located in the cockpit. Details of the IPL Control Logic and the associated hardware tests performed are explained in section 9 and 10 of the Systems Brief, GPC Initial Program Load. As a result of the hardware/firmware response to the IPL Control Logic, the GPCIPL software program loads the PASS or BFS resident software and data areas into the GPC from the MMU. An illustration of the PASS IPL load process is shown in figure 11.3.3-1. After the GPC successfully completes all of its self tests and the system software is loaded, the Software System Loader (SSL) or Backup System Loader (BSL) gives control of the GPC to the PASS or BFS. If control is given to the BFS, FCOS involvement terminates. If control is given to the PASS, the PASS resident software completes the initialization. The PASS_GPC_ Initialization_Processor (FCMLINIT) initializes the system software for a PASS GPC. A flow diagram of the process is shown in
figure 11.3.3-2.

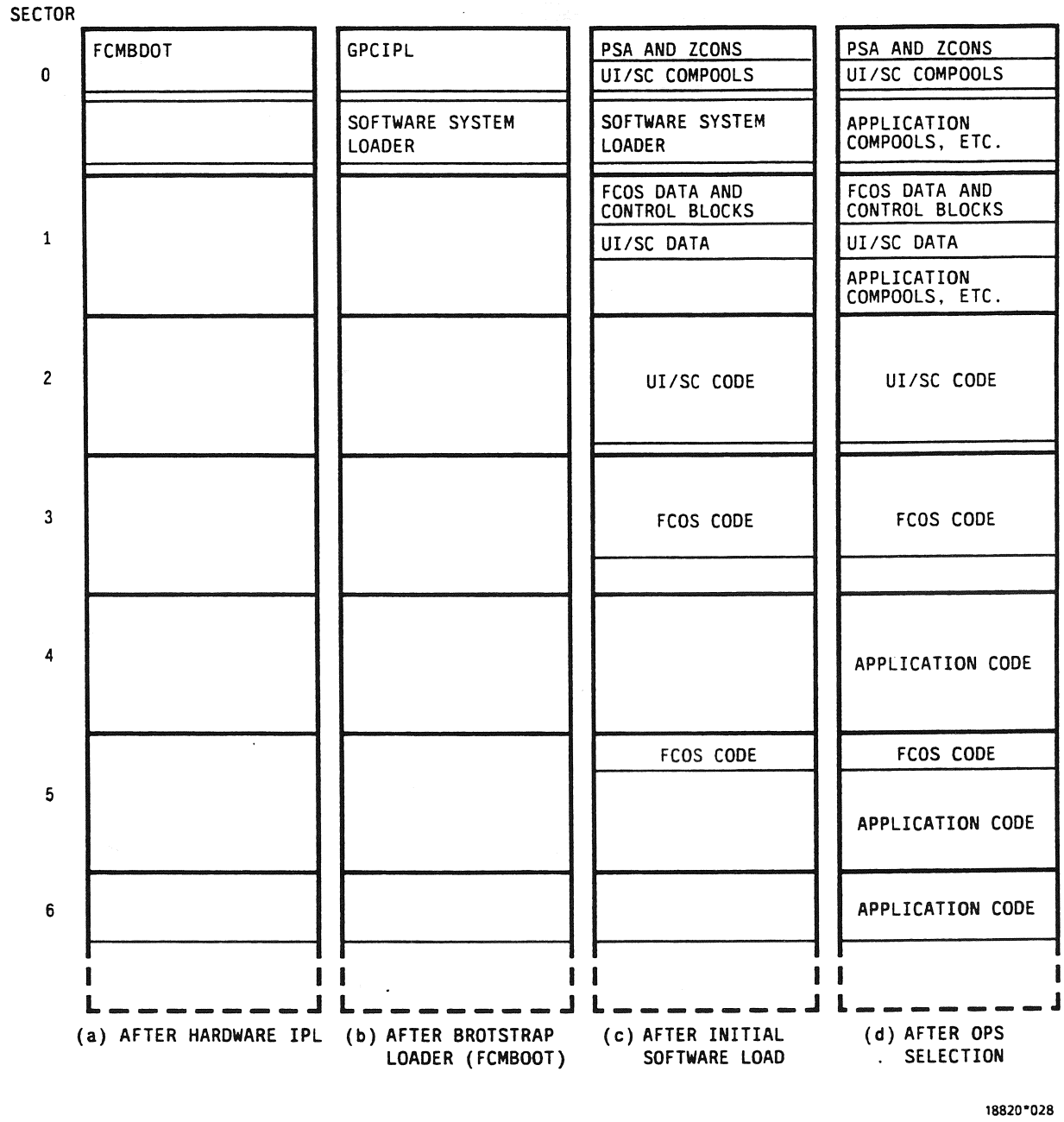


Figure 11.3.3-1.- Primary Avionics Software System IPL.

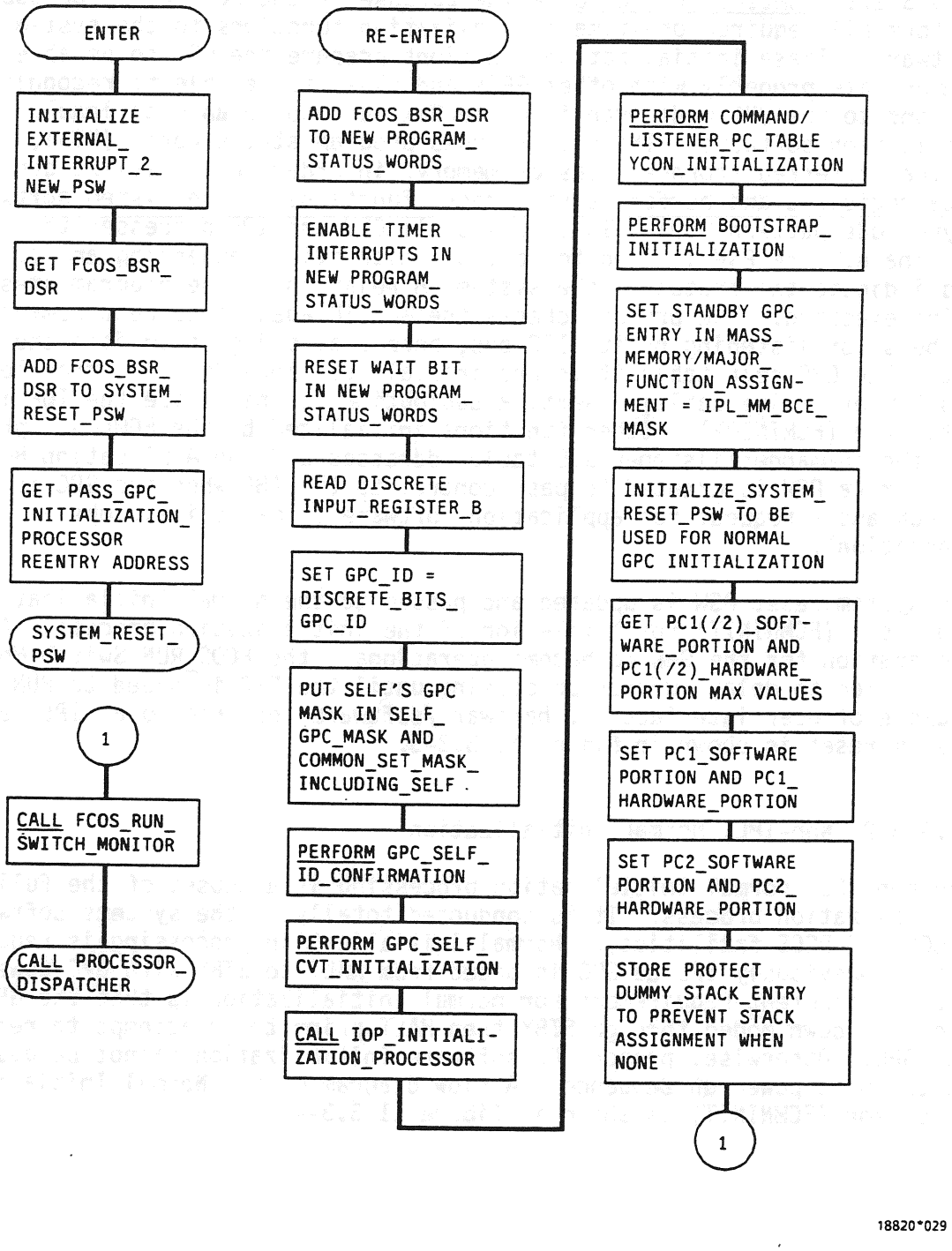


Figure 11.3.3-2.- Pass_GPC_Initialization_Processor (FCMLINIT).

11.3.3.1.1 FCMLINIT Processor.- The purpose of the FCMLINIT processor is to perform all required one-time initialization functions to the system software. These initialization functions prepare the GPC to be able to communicate properly with other GPCs and LRUs, to be able to recognize and respond to the GPC mode transition to RUN, and to be able to load application software. The program uses program status words (PSW), located in the preferred storage areas of memory, in order to initialize general data registers and perform system reset functions to the system software (sync code pattern 000). This causes all CPU and IOP processes to terminate. The PSW is used to control instruction sequencing and to hold and indicate the status of the system in relation to the program presently being executed. The program checks the GPC ID against current common set members, by listening on its ICC bus, before allowing it to join the common set. The GPC self table items are initialized (includes I/O error log table and GPC error log table). Next, a subroutine to initialize the IOP is activated (FCMINIOP). Other functions initialized by the FCMLINIT processor are the commander/listener BCE table addresses and the Application Bootstrap Program (a PSW is created to pass control to the ABP when the GPC is moded to RUN and a request for application software is made; i.e., ops transition).

The system reset PSW is updated and passed to the Normal Initialization processor (FCMNINIT) for completion of the initialization process. As final preparation for the GPC to become operational, the FCOS_RUN_Switch_Monitor is invoked to delay further processing until the GPC is moded to RUN. An example of user interface vs. hardware/software response to an IPL sequence system reset is shown in figure 11.3.3-3.

11.3.3.2 Non-IPL, NormaL Initialization

The non-IPL, normal initialization processing is a subset of the full IPL initialization process. It is conducted totally by the systems software (FCOS and FCOS facilities). Normal initialization processing is required when a previously IPL'ed GPC is moded from HALT to STBY with GPC power ON. Another inherent requirement for normal initialization is that the GPC be properly down moded through STBY then HALT prior to an attempt to reactivate the GPC. Otherwise, proper GPC software initialization cannot be guaranteed at the next power up sequence. A flow diagram of the Normal Initialization processor (FCMNINIT) is shown in figure 11.3.3-4.

#	USER ACTION	PANEL	GPC HARDWARE	GPC SOFTWARE
1	Mass Memory Power on - MMU1 MMU2	014 015	No Response	No Response
2	DEU(s) Power -STANDBY	C2A2	No Response	No Response
3	Mass Memory IPL Source Select to desired MMU	O6	No Response	No Response
4	GPC to HALT mode	O6	No Response	No Response
5	GPC Power - ON	O6	No Response	No Response
6	BFC CRT display switch to ON (Menu IPL only)	C3A1	No Response	No Response
7	BFC CRT SELECT switch to 1+2, 2+3, or 3+1 (Menu IPL only)	C3A1	No Response	No Response
8	DEU(s) Power - ON (at least 30 seconds after step 2)	C2A2	No Response	No Response
9	DEU LOAD - Push, then release (P/R) (Menu IPL only)	O6	No Response	No Response
10	GPC IPL - P/R (at least 2 minutes after step 1)	O6	Drives mode TB-IPL; loads fixed pattern of C9FB in memory (0-1FFFF); loads bootstrap loader from MMU; resets mode TB Barberpole	No Response

Figure 11.3.3-3.- Default IPL Sequence.

#	USER ACTION	PANEL	GPC HARDWARE	GPC SOFTWARE
11	GPC to STBY mode	O6	STBY discrete (D101)-ON.	Bootstrap fetches GPCIPL and SSL from MMU and passes CPU control to GPCIPL. STP executes one cycle; if successful, GPCIPL gives control to SSL. If step 6 is done, GPCIPL loads DEU, selected in step 7, from MMU; IPL menu presented. (Go to step 12.) If step 6 is not done, SSL loads PASS area 1 phase 2 into memory; mode TB-RUN. (Go to step 13.)
12	Select system to be loaded (Menu IPL only) 1 PASS Area 1 3 PASS Area 2 5 PASS Area 3 2 BFS Area 1 4 BFS Area 2 6 BFS Area 3	KYBD	No Response	If PASS selected, system phase 2 loaded, Mode TB-RUN. If BFS selected, the BSL21PL menu will be presented. (Go to step 14.)

Figure 11.3.3-3.- Continued

#	USER ACTION	PANEL	GPC HARDWARE	GPC SOFTWARE
13	GPC mode - RUN	O6	RUN discrete (D102)-ON	Enters OPS 0, presents GPC Memory display. Note: To prevent ICC conflicts, wait 10 seconds after moding to RUN before doing any operations listed under the GPC mode discussion.
14	Mass Memory - IPL Source Select OFF	O6	No Response	Removes mask to allow access of MMU. For BFS, present GPC Memory on CRT.

Figure 11.3.3-3.- Concluded

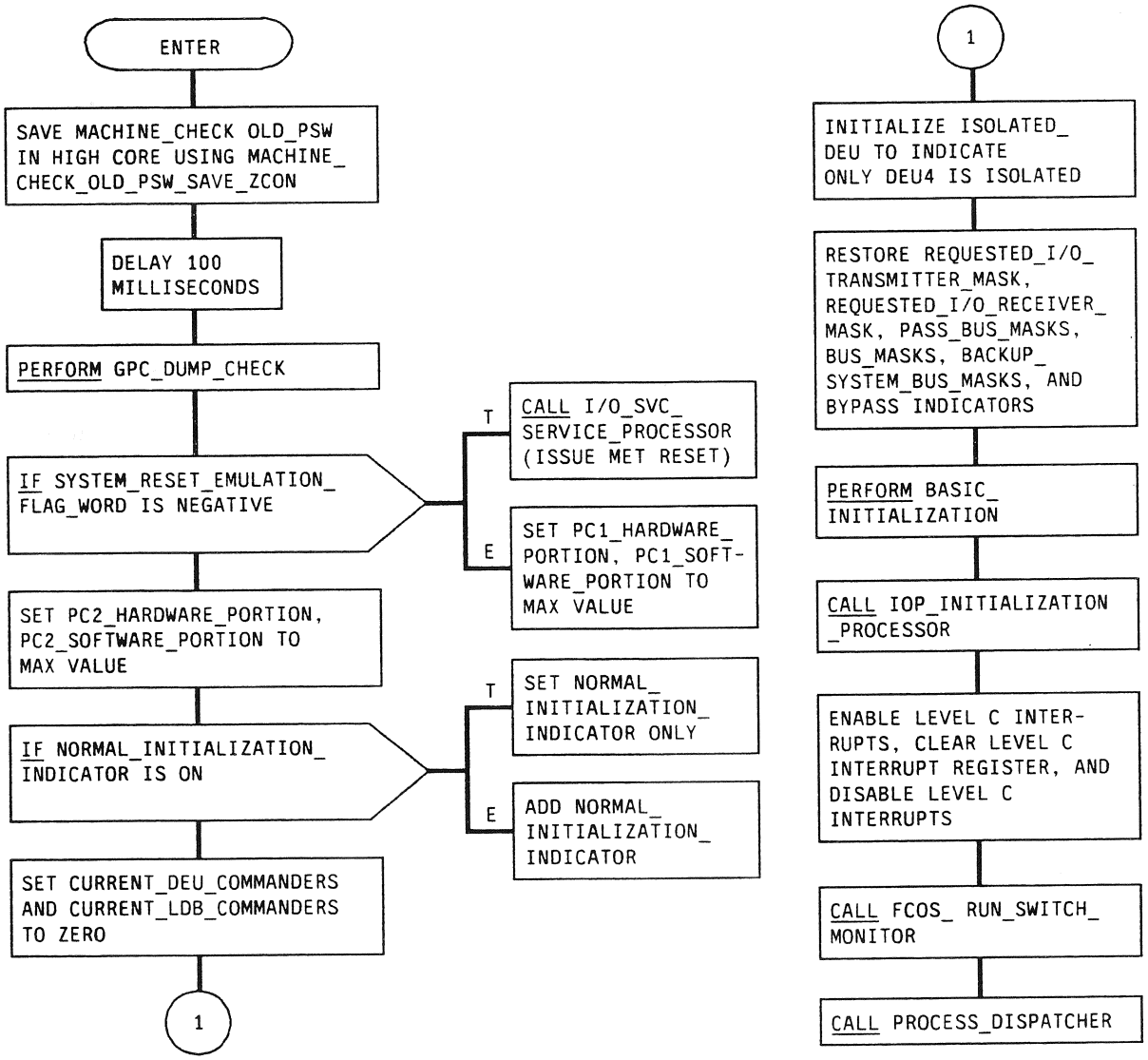


Figure 11.3.3-4.- Normal_Initialization_Processor (FCMNINIT).

11.3.3.2.1 FCMNINIT processor.- The normal initialization process performs various system reset functions and several ancillary functions. The first function performed is to ensure that the old machine check PSW is saved in the proper area (sector 6) so that it will not be written over by system or application software. The PSW is updated periodically and used as an indicator to the software that the proper hardware and software checks have been performed and/or initialized. Next, a 2-second delay is incorporated in the software for human erratic switch movement (OI8-C). The delay causes the GPC to delay its call to the IOP initialization processor until the discretes have settled. This delay may be removed from the code later if it is found that no GPC still has this problem.

An ancillary function performed by this processor is to read the Discrete Register A to check for a dump request (normal initialization is performed for a GPC properly downmoded through STBY, then HALT, after an apparent failure). If the STBY, I/O TERM B, and dump discrete bits are all set, then the GPC stand-alone dump is invoked which configures the GPC for a HISAM dump and places the GPC in a wait state (no other initialization functions are attempted). Dump configuration functions include reset of the IOP, enabling BCE 24 transmitter and receiver, configuring the MSC and BCE 24, and setting the necessary flags for the dump.

If no dump request is detected, the initialization process is continued. Various system reset flags/indicators are set. The application bootstrap program (AIB) is initialized at this point. Mass Memory active and I/O queues are cleared. All BCE bypass indicators are reinitialized along with various other synchronization and time-related data. The IOP initialization processor (FCMINIOP) is then activated which initializes the required I/O processes, the flight critical buses, and the BCE timeout registers. The GPC fail light is reset at this point. All level C interrupts are cleared and the system software is then placed in a cyclic loop until the GPC is moded to RUN.

11.3.3.2.2 Completion of GPC initialization.- Once a GPC has successfully completed initialization in the STBY mode, software places the GPC in a cyclic loop to await transition to the RUN mode. Upon completion of the required tests and checks, a confidence level for the GPC should have been acheived by the user. This confidence level is reflected in applicable error conditions and/or messages generated from the hardware BITE or system software. Once a confidence level has been attained and a user modes the GPC from STBY to RUN, the FCOS (as another step in completing the initialization process) enters the GPC in the first level of computer synchronization required to support multi-flight computer operations. This first level of synchronization is called common set synchronization; and it is discussed in detail in section 11.3.1 of the Systems Brief, GPC Common and Redundant Set Synchronization. The following information only serves as a brief introduction to the subject matter and establishes its correlation with GPC initialization.

11.3.3.3 System Initialization

There is an area of systems software called System Control which performs system services that are neither a function of FCOS nor a function of User Interface. These functions include initialization of the system following the loading of a GPC (via IPL or other means), support for OPS 0 functions, system reconfiguration control and implementation of a set of specialist functions which are systems oriented rather than directly related to a major function. A functional overview heirarchy diagram is shown in figure 11.3.3-5.

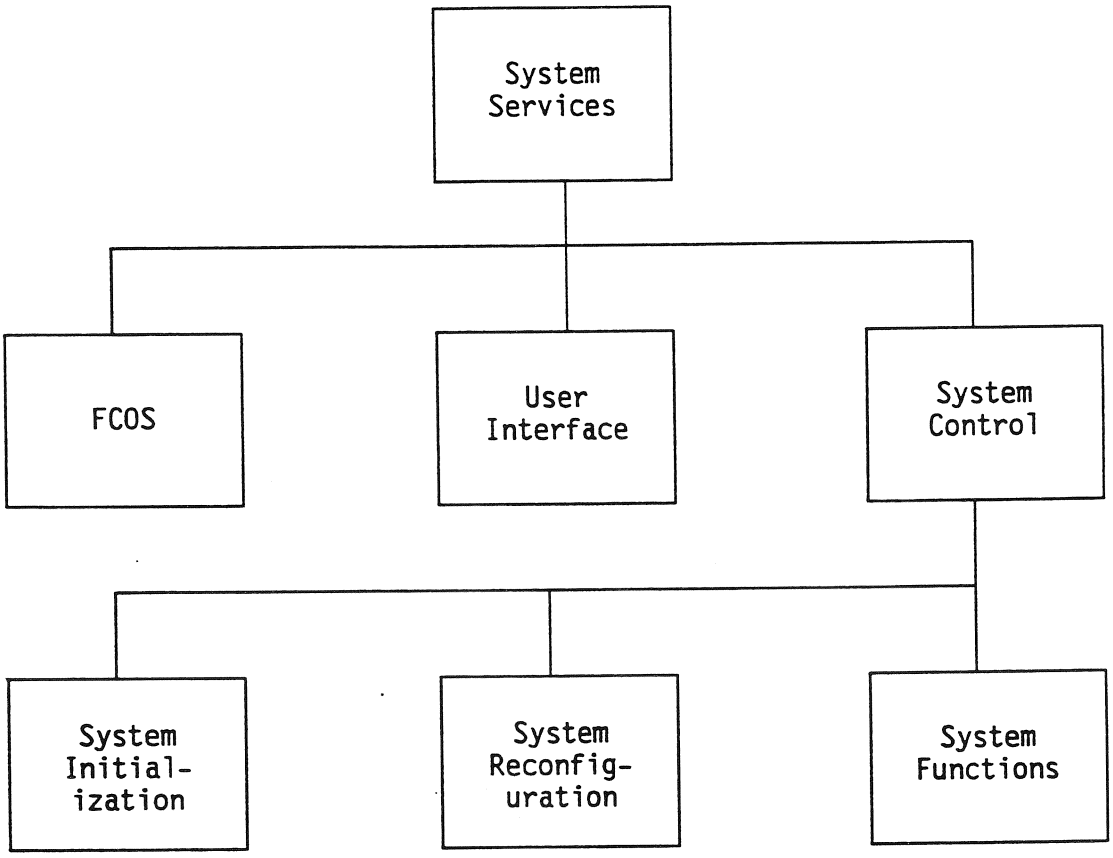


Figure 11.3.3-5.- Functional Overview Hierarchy Diagram.

The Systems Initialization software initializes the execution of user interface and systems control software, and places the GPC into a state from which normal operation may begin. The software is designed to start only one GPC at a time. In the event where a redundant set of GPCs is to be initialized, the initial GPC goes through what is called a primary GPC initialization; while the subsequent GPCs are placed in a secondary mode. All GPCs activated are synchronized at a common synch point.

Initial Start Sequence software provides the interface with FCOS to initialize GPC operations. It differentiates the first GPC from the secondary GPCs and executes the startup algorithm accordingly. If successful, the READY talkback is turned on following the initiation of other system services software.

11.3.3.3.1 AIB_GPC_LOCATOR.- The software module which controls the initial GPC interface function is called AIB_GPC_LOCATOR. The functions of this module are to initiate the GPC execution by identifying the common GPC set, to initialize the internal clock, to schedule the necessary processes for establishing user interface, to initiate GPC OPS 0 and to turn on the GPC ready indicator. A pseudo schedule is created by FCOS by obtaining the PDE for the GPC_LOCATOR at a predetermined location. FCOS Normal Initialization routine creates the schedule.

The initial SSIP SYNC SVC is issued to test for the existence of a common set. Then a series of compool data items are initialized and various events are reset. Then the ICC and PMU busses are enabled. If there is not a common set, then primary GPC initialization is performed. Otherwise, secondary GPC initialization is performed.

Primary GPC initialization starts with more data items being initialized. The four strings and buses 6 to 13 (excluding 9) are enabled. The GPC internal clock is initialized, the toggle buffer is assigned, and downlist I/O is initiated. Annunciation processing is cleared and LDB processing is initialized according to its previous state.

Secondary GPC initialization starts with the four strings and buses 6 to 13 being disabled. Various other functions common to primary initialization are performed at this point. If the current common set commands no DEU, then DEUs 1 to 3 are assigned to the new common set GPC. Now, GPC OPS 0 is initiated, the processes necessary for establishing user interface are scheduled, and the GPC ready indicator is turned ON.

11.3.3.3.2 GPC internal clock initialization.- Following the transition of any GPC from HALT-STANDBY-RUN or STBY-RUN mode, as a part of the initialization process, a GPC's internal timers are synchronized with the Master Timing Unit (MTU). For all practical purposes, the GPC's internal time source (RUNTIME) is initialized to be equal to the RUNTIME of the lowest GPC ID in control of flight-critical buses 1 to 3 or at least one of the FC 1 to 3 buses. Thus, after transition to RUN, the GPC initiates its time management processing (TMP) for common set sync (SSIP) using this same time source. However, if no other GPC is active (this is the first GPC activated), the GPC will initialize its RUNTIME with its first available

time source (MTU accumulator 1, 2, or 3 respectively). This time is passed to TMP as the initial RUNTIME (which can cause a PMU TIME INVALID GPC error when compared against a PCMMU-generated time source for "reasonableness" during redundancy management). An operational workaround has been developed, per User's Note DR100710, forcing the MTU accumulator time to be within "reasonable" limits of the PCMMU-generated time. The MTU time is then used as the RUNTIME. In addition, as a part of the initial startup procedure, a validity check is performed on GMT and MET source data, prior to accepting an MTU accumulator as a time source. Failure to fall within fixed limits (days:1-399, hours:0-23, min:0-59, sec:0-59) will cause the MET to be reset to zero and RUNTIME to be set within limits.

11.3.3.4 References

1. NSTS Computer Program Development Specification (CPDS) NSTS System Level A Software Requirements, SS-P-002-170K, September 1, 1989.

2. Space Shuttle Programs Orbiter Avionics Software Operational (OPS) Detailed Design Specification (DDS), Volume: II-System Services, Part 1 - Flight Computer Operating System, Prepared by IBM under NAS 9-14444, April 2, 1982.

3. DPS Systems Brief, IPL-AP-101B, CS/RS Sync, Basic, Rev B, JSC-18820, June 30, 1989.

4. System Software fische (FCOS), DASS DISASSM-SSW.

SECTION 12
PASS INPUT/OUTPUT

CONTENTS

Section		Page
12.1	INTRODUCTION	12-1
12.2	GPC INTERNAL OPERATIONS	12-1
12.3	DATA BUS NETWORK	12-2
12.4	BYPASS CONDITIONS	12-7
12.5	I/O ERROR CONDITIONS	12-8
12.6	I/O RESET	12-10
12.7	MDM OUTPUT CONDITIONS	12-11
12.8	FCOS I/O ERRORS	12-12
12.9	RECENT STUDIES	12-12
12.10	REFERENCES	12-13

SECTION 12
PASS INPUT/OUTPUT[1,2]

12.1 INTRODUCTION

In multi-GPC set operations, it is essential to provide special handling and monitoring of input data transactions (known as protected transactions) in order to establish and maintain a common input data set. In addition, for either single or multi-GPC operations, non-transient failure conditions need to be bypassed in order to eliminate any steady-state CPU overhead that would otherwise occur from a continuous I/O error processing situation.

However, output transaction failures are generally transparent to the GPC(s), that is, within the constraints established by bus assignments (bus commander versus listener). All output transactions are executed by the IOP without regard to their eventual success or failure. There is no requirement to eliminate or isolate error sources, and only certain output transaction errors are annunciated. Error logging of non-transient output failures shall cease after two consecutive errors on the I/O transaction; the assumption being that a permanent output error condition exists. I/O error monitoring, therefore, is essentially limited to input transactions.

Each GPC maintains a log of data representing the five most recent I/O transaction and IOP hardware errors. Through ICC transactions, each GPC also maintains a log of data for all GPCs in the common set. This log consists of (1) the most recent (last) I/O error entry and (2) the cumulative count of all I/O errors detected by each common set GPC. In addition, each GPC maintains a log of all input BCE elements and BTU segments bypassed within the active GPC set. Such data are downlisted in the form of BCE bypass words.

12.2 GPC INTERNAL OPERATIONS

Internal to the GPC, it is convenient to separate the I/O function and look at it from two sides: the CPU side and the IOP side. On the CPU side, system and application software process execution is based on a major cycle, which is 960 milliseconds long or runs at 1.04 Hz. A major cycle is broken up into 24 minor cycles, each 40 milliseconds in length. This minor cycle architecture allows for software processes to execute at rates up to 25 Hz. (The 25 Hz rate is required to provide the command and control needed for the GNC flight control software, especially in the dynamic phases of ascent and entry.) A minor cycle can also be partitioned into input, process, output, and other sections. This entire scheme is controlled by various CPU hardware interrupts.

The low-, mid- and high-frequency executives (LFE, MFE and HFE) are GNC software processes that schedule and separate the GNC application software processes and I/O into those with rates of up to 1.04 Hz (once every major cycle), 6.25 Hz (once every fourth minor cycle) and 25 Hz (once every minor cycle), respectively. Although these executives are implemented only in the

GNC software, their rate divisions are also used for the SM and PL applications software. The SM and PL software, however, have no software running at HFE rates.

The G9 software has been written somewhat differently, because of its use during the prelaunch/postlanding timeframes. There are no HFE processes, but the 25 Hz Housekeeping Data Acquisition (HDA) processes and I/O take their place. Also, there are no MFE processes in G9 or any processes that take their place, but MFE I/O does take place at a rate of 12.5 Hz.

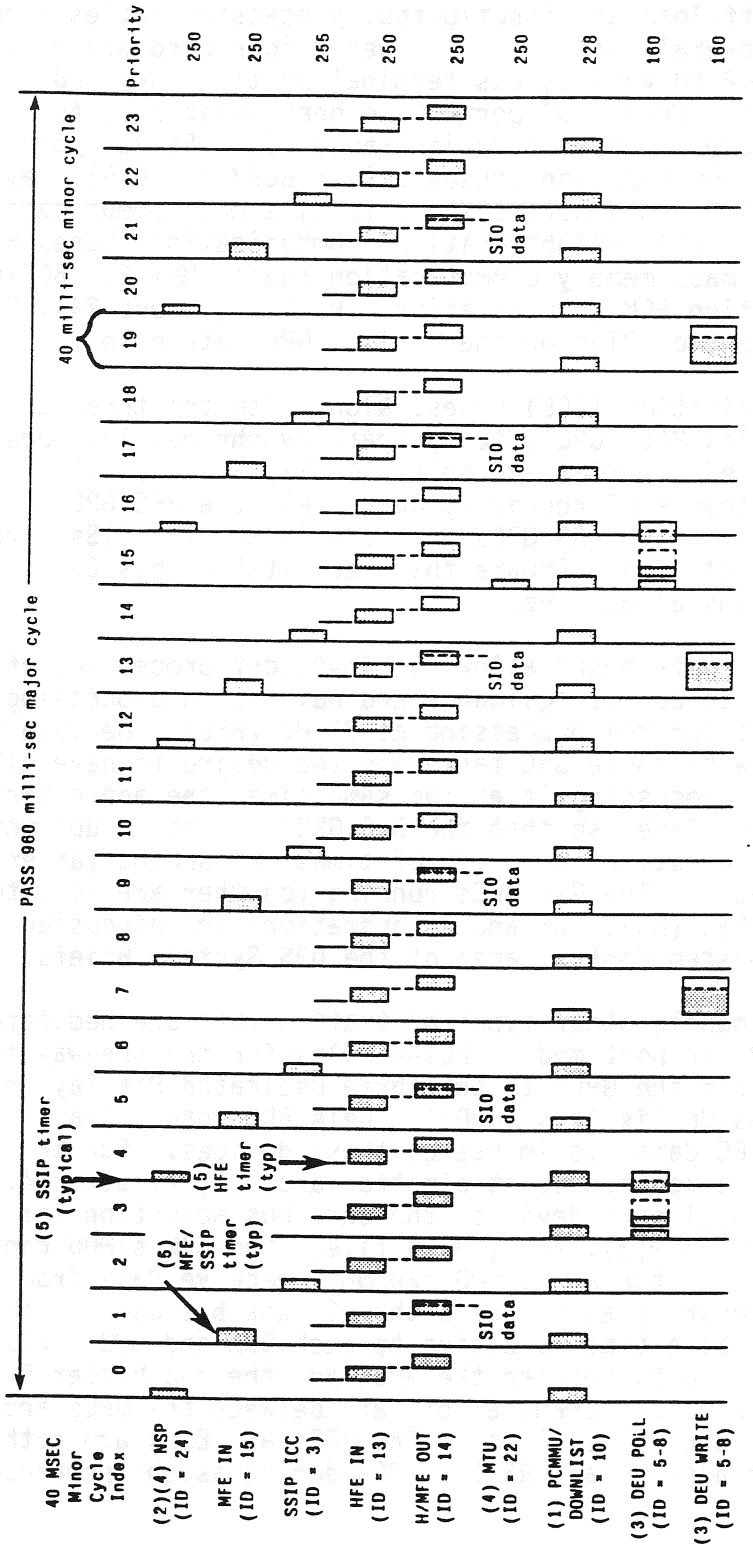
A skewing of the input/output processes (I/O profile) in a minor cycle due to I/O error, FTS, or other higher priority processing can cause an HFE, or much less likely, an MFE, cycle overrun (commonly called a cycle wrap) GPC error to be logged. HFE cycle wraps should cause no long-term adverse affects on software performance, but for an MFE cycle wrap these long-term adverse affects could occur. Table 12-I is shows an overview of nominal PASS GNC I/O.

On the IOP side, one can break up the I/O scheme into the following three levels: (1) program controlled I/O (PCI/PCO) and direct memory access (DMA), (2) the master sequence controller (MSC), and (3) the bus control element (BCE). At the top level, the PCI/PCO operations are CPU initiated and controlled; whereas, the DMA operations are IOP initiated and controlled. The PCI/PCO instructions are used to direct the IOP MSC, while the DMA operation allow GPC memory access without CPU knowledge or intervention. A DMA operation has priority over a PCI/PCO operation should a simultaneous access occur. A backlog of DMA traffic will cause the DMA to go into a "burst" mode to relieve its load, thus temporarily suspending PCI/PCO operations.

At the next level, the MSC receives PCI/PCO instructions, directs the time-shared activities of the IOP BCEs and notifies the CPU of I/O completion. At the lowest level, each of the 24 IOP data buses is associated with a BCE program. BCE program execution is coordinated by the MSC. Each time slice (or microcycle) allocated by the MSC for each BCE program to execute lasts 0.5 microsecond, and the MSC takes every fourth time slice to coordinate the I/O activity as necessary. There are 33 microcycles (lasting 16.5 microseconds) allocated to allow each BCE and the MSC the time to cyclically execute.

When each BCE program is executed, it first checks the status of its associated Multiplexer Interface Adapter (MIA). If the MIA is not busy, and it normally will not be busy, the I/O transaction is executed. The response to a busy MIA is different for input transaction elements versus output transaction elements. An input element will cause no action to be taken, and the MIA will be checked again during the BCE's next alloted time slice (in 16.5 microseconds). An output element will react to a busy MIA with an I/O error. (On the AP-101S GPC, a busy MIA on an input element will cause an I/O error to be logged just as for output transactions. This will decrease the likelihood of continuous I/O errors and BCE bypass fault messages from occurring for failure modes such as the loquacious MDM.)

TABLE 12-I.- PASS GNC "NOMINAL" I/O TRANSACTION OVERVIEW



18820*034

Notes: (1) Downlist is the only PASS I/O transaction not supported (supervised) by MSC monitoring.
A dump replaces the normal downlist data stream; the rest of software timing is unaffected.
(2) 'NSP POWER ON' discretes are monitored to detect an NSP switch, and therefore an FC/NSP data bus switch.
(3) DEU cyclic I/O is not timer initiated. DEU poll software (display MCDS input processor - DMI) module scheduled at 2.08 Hz rate (twice per major cycle) with priority = 232. I/O initiated via I/O svc no. 24. DEU write (fill) software (cyclic display processor - DCICYC) module scheduled at 2.08 Hz rate with priority = 153. I/O initiated via I/O SVC no. 24. Consequently, DK bus I/O activity will vary considerably, depending upon both CPU and IOP real-time processing.
(4) At OPS transitions, only NSP and MTU transactions continue; the rest are suspended.
(5) SSIP timer interrupt defines start of each 40 milli-sec minor cycle (PASS only).
HFE timer interrupt occurs 14 milli-sec after SIP timer interrupt.
MFE does not have a unique timer interrupt, because the MFE process and I/O are kicked off by the SSIP timer interrupt on the specified SSIP cycle.

12.3 DATA BUS NETWORK

The GPC has an IOP to off-load the input/output processing duties from its CPU, thereby improving overall efficiency. Twenty-four hardware ports and buses connect the GPC IOP to various bus terminal units (BTUs) and line replaceable units (LRUs). These IOP ports have been designated for the following uses: intercomputer communication (buses ICC1-5: BCE/port 1-5), display/keyboard communication (buses DK1-4: BCE/port 6-9), payload communication (buses PL1-2: BCE/port 10-11), (pre)launch communication (buses LB1-2: BCE/port 12-13, flight-critical communication (buses FC1-8: BCE/port 20-23, 14-17), mass memory communication (buses MMI-2: BCE/port 18-19), and instrumentation PCM communication (IP1 [2-5]: bus 24). Figure 12-1 shows the arrangement of BTUs on the various GPC data buses.

The Intercomputer Communication (ICC) buses, along with the three GPC discrete sync lines, allow PASS GPC's to operate synchronously. One ICC bus transmitter is enabled per PASS GPC (e.g., GPC2/ICC2), but all PASS GPC ICC bus receivers are capable of receiving data. All the BFS GPC ICC transmitters are disabled. All PASS GPCs that are in RUN (no FTSs) are said to be in a Common Set (CS). The software that facilitates this CS communication process runs at 6.25 Hz.

The Flight-Critical (FC) data buses allow for in-flight processing of GNC information at rates up to 25 Hz, guidance and navigation processing at 6.25 Hz (MFE) and flight control processing at 25 Hz (HFE). Because of the FC processing involved with these GNC tasks and the desire to have all GNC GPCs performing I/O and processing it at the same time, the above three GPC sync lines are further utilized so that the GNC GPCs can "sync up" more tightly than in a CS at a rate of hundreds of times per second (after each I/O transaction, SVC, etc). The GNC GPCs running together are said to be running in a Redundant Set (RS). CS and RS operations are discussed in greater detail in the System Control area of the DPS Systems Briefs.

The FC data bases also handle other types of traffic that are dedicated, i.e., cannot be changed via port mode. FC1-4 allow for the one-way transfer of data (output only) from the GPCs to the three Dedicated Display Units (DDUs) and the two Heads-Up Displays (HUDs). DATA BUS rotary dials for the CDR and PLT select the FC data bus in use by these devices. For the quad-port DDUs, the data bus selection is straightforward: 1, 2, 3, or 4. But for the HUDs which are dual port devices, the data bus selection is: CDR - 1, 2, 3(1), 4(2) and PLT - 1(3), 2(4), 3, 4 (i.e., the CDR's HUD can only recieve data from FC1 or 2; the PLT's HUD can only receive data from FC3 or 4). The aft DATA BUS rotary dial controls the FC data bus used by the aft DDU. Only one data bus at a time is active to each DDU and HUD. FC5-8 allow for the transfer of data between the GPCs and the two Master Events Controllers (MECs) and also the transfer of data between the GPCs and the three SSME Engine Interface Units (EIUs). The MECs and EIUs are both quad-port devices with interfaces to all RS GNC GPCs during ascent powered flight.

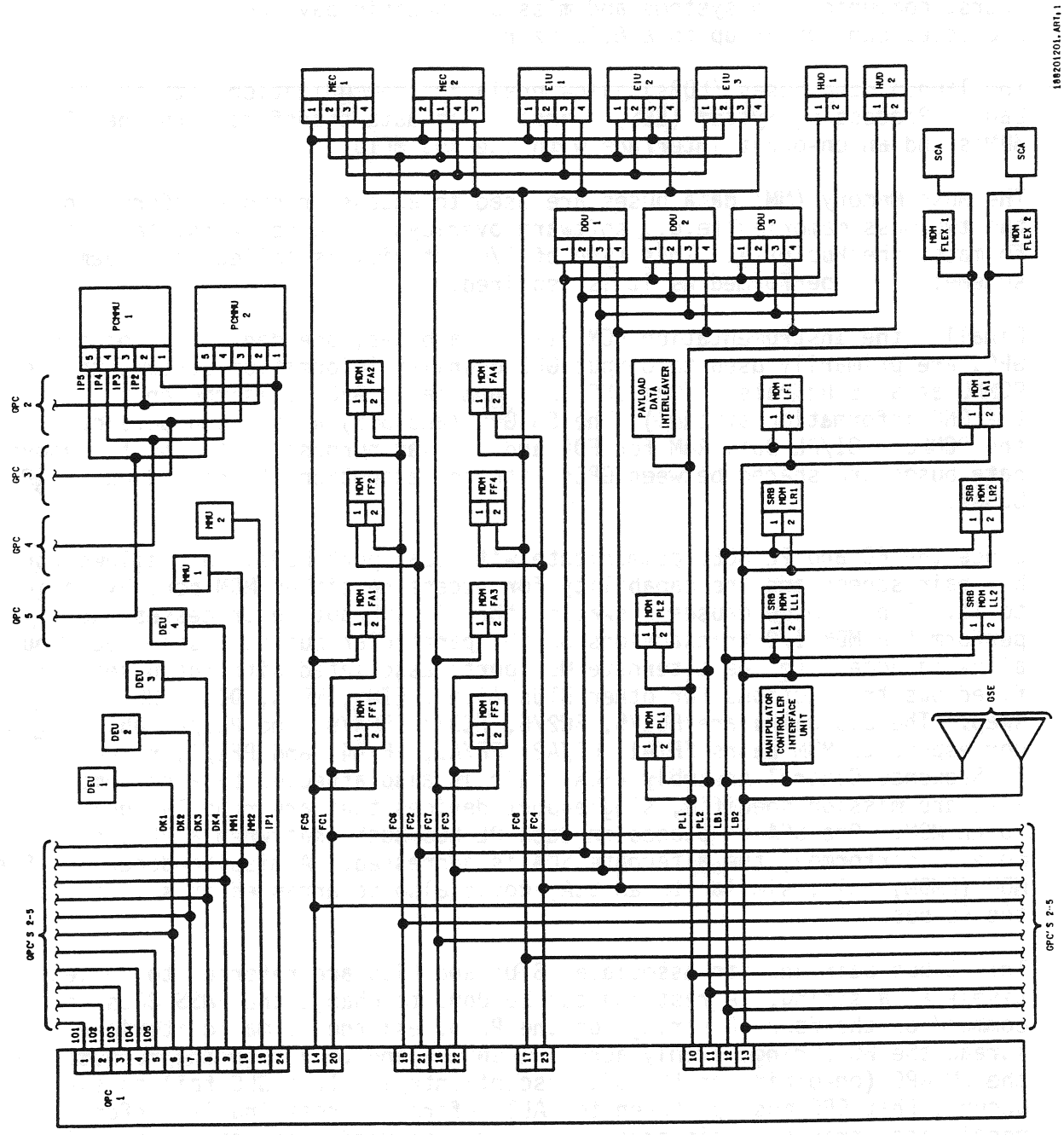


Figure 12-1.- DPS data bus schematic.

The Display/Keyboard (DK) buses allow any GPC to communicate with any of the MCDS DEUs which are polled at a 2.08 Hz rate.

The Payload (PL) buses give the GPC an interface to control the payload bay doors, communication systems and mission specific payloads. The payload processes can run at up to a 6.25 Hz rate.

The launch data buses (LDBs) allow prelaunch communication with the KSC Launch Processing System (LPS), first stage data transfers with the SRB MDM's and an on-orbit interface with the RMS MCIU.

The Mass Memory (MM) data buses are used to access or store information on the two mass memories (e.g., software overlays, SPEC roll-ins, telemetry formats, checkpoints). This type of I/O activity is called an on-demand scheme; i.e., performed as it is required.

Finally, the Instrumentation PCM (IP) data buses, one dedicated for each GPC, are primarily used to output GPC downlist information streams to the PCMMU at a 25 Hz rate. (One GNC GPC, nominally GPC 1, is chosen to downlist the GNC information stream.) The SM GPC (and BFS) also fetch data out of the PCMMU's OI/PL Data RAM for FDA and display purposes. All of the above data buses are shared between GPCs with the exception of the dedicated IP buses.

Since the FC and PL BCEs communicate with dual port MDMs, this allows for a bus pair scheme and the capability for accessing either MDM port by moding to it. A port mode causes a swap of the BCE subroutine processes that perform the MDM I/O transactions on the particular bus pair selected, thus allowing access to the alternate MDM ports associated with the buses. The fixed bus transactions for other BTUs (i.e., EIUs, MECs, DDUs, HUDs) do not swap. The bus pairs are FC1/5, FC2/6, FC3/7, FC4/8, and PL1/2, which correspond to MDM pairs FF/A1, FF/A2, FF/A3, FF/A4, and PF1/2, respectively. (A Sequence Control Assembly (SCA) pair is also affected by a port mode. The SCAs are mission-specific, single-port devices that are roughly equivalent to an MDM. One SCA is placed on each PL bus such that when a PL bus port mode is performed, the alternate SCA is addressed. A mission-dependent Flex MDM (FMDM) pair, similar to an SCA, could also be arranged this same way on the PL buses.)

The FC bus pair and its associated BTUs and LRUs are referred to collectively as a string. A restring can be done to change the PASS GPCs in command of the four FC strings or the PL buses; the normal criteria being to spread the FC strings evenly across a GNC RS and to assign the PL buses to the SM GPC (on-orbit) or the BFS (ascent/entry). If a GPC fail-to-sync occurs, that GPC must be taken to HALT before any restring is performed to regain its strings. This precludes a dual-commander situation, two GPCs transmitting on the same data bus, from causing further RS problems. A string imbalance of two or more between any two RS GPCs in an RS of three or more GPC's can lead to a fail-to-sync (FTS) at an SVC sync interrupt on the GPC with the high string overload.

Input I/O errors logged by a GPC can eventually lead to either of two types of fault messages being annunciated: bypass messages or I/O ERROR messages. The logging of a "single hit" I/O error on a bus should not be confused with the annunciation of an error message on the fault summary page (SPEC 99). GPC outputs are not affected as a result of the aforementioned input error annunciations. However, the FC and PF MDMs have a wraparound test that will cause an MDM OUTPUT XXX annunciation if a output failure is detected. Each of these error types will now be described in more detail.

12.4 BYPASS CONDITIONS

Each BCE program executes a series of subroutine processes; the resulting I/O activity being returned to the GPC in the form of transactions made up of BCE elements. Although some I/O transactions are not bypassable (e.g., ICC, DK, MM, and IP/downlist), the following scheme is used for the flight-critical, payload, and other transactions. A single I/O error on input data from any element in a transaction will cause a commfault to be logged (a bit set to one) for every element/LRU on that transaction.

If the next (consecutive) transaction is error free, the commfaults are removed, i.e., the appropriate bits are reset to zero. If the next (consecutive) transaction also yields an I/O error, a bypass message will be annunciated and the element at fault will be unchained from the I/O transaction. The appropriate I/O RESET command rechains bypassed elements. If the error condition is hard; i.e., not transient, the bypass will be reestablished. Turning off an LRU will normally yield BCE SEV errors and a bypass message, whereas consecutive I/O errors on a PROM sequence will have (non-initial) BCE timeout signatures and yield a bypass message.

SEV errors can be described as follows: an "S" bit reset is caused by a power transient detected by the MDM, an "E" bit set is caused by a MDM serial I/O card detecting a data problem, and a "V" bit reset is caused by an internally-detected MDM data problem. The SEV bit pattern is normally 101 and is contained in each status register for BCEs 1 through 24.

A few statements about MDM input transactions and their BCE elements need to be understood. An MDM has two types of input transaction elements: direct read elements and indirect read elements. The direct I/O elements are associated with serial data buses to individual LRUs (e.g. an IMU, a NSP, the MTU), while the indirect I/O elements are implemented via MDM PROM sequences (A, B, C and D) which gather various analog and discrete data. A problem with either a direct or indirect element in an input transaction can yield a bypass message of the form BCE STRG X YYY, where X is the string number and YYY is the LRU or PROM sequence that was bypassed. The reason that serial interfaces were not placed on the PROM sequence elements and were given their own individual elements is, if an LRU had been turned off with the former scheme implemented, it would have caused the entire sequence (analogs, discretes, and other serial data) to be bypassed.

The key point to remember about bypass conditions is they indicate only a loss of input communication from a particular chained BCE element, i.e., a

non-initial (not the first on the bus) transaction element timeout or SEV error has occurred, and they occur only after two consecutive I/O errors. Again, outputs (commands) are not affected by bypass situations on input transactions.

Many BCE elements are application/OPS dependent (e.g., GNC HFE and MFE elements), and the bypass logic is inhibited during OPS transitions while new BCE element tables are being established. However, FCOS-implemented transactions continue (e.g., MTU, NSP, downlist). Should a failure occur in one of the noted FCOS areas, I/O errors would be logged at the transaction execution rate until the transition is completed.

SM software utilizes a separate set of IP bus I/O elements and bypasses for the PCMMU and its OI, PDI, and Spacelab interfaces. This arrangement was implemented to facilitate ground as well as crew control of the vital PCMMU data link. Upmoding of these I/O elements is accomplished using a crew input Item 5 (I/O RESET PCM) on SM SPEC 62 or an equivalent uplink command via a dedicated op-code. A general SM I/O RESET EXEC will accomplish the same upmoding of the PCMMU I/O elements, but will potentially also annunciate nuisance alarms from the re-bypass of I/O elements on other buses. SM also has bypass logic implemented for mission-specific BTU's (PAM SCAs, Flex MDMs, etc.) on the PL buses, whose elements can be upmoded using a general SM I/O RESET EXEC. Consult the SM Systems Brief for details on the OI MDM, PDI and Spacelab I/O interfaces.

This I/O scheme becomes much more complicated in a redundant set of GPCs because of the synchronization that must be performed between them. The I/O error data is ICC'd so that an assessment can be made (on the second consecutive error) as to which GPC(s) saw the error and which BCE element was at fault. The simplest case is that all the RS GPC's see the same BCE element at fault and it is bypassed with the appropriate bypass fault message annunciated. This is called a universally detected I/O error. If a single GPC sees the error, this is considered a non-universal I/O error, the BCE element is not bypassed, and this GPC will force FTS.

If more than one, but not all, of the RS GPCs see the same error (e.g., ascent - two or three GNC GPCs), the BCE element in error is bypassed. This is not a non-universal I/O error because more than one GPC has detected it, but it is not a universally detected I/O error either. If the RS GPCs perceive that a different BCE element is at fault, the highest numbered element in the transaction; i.e. the one closest to the end of the BCE chain, is the BCE element that is bypassed. The theory behind this methodology is that eventually all the errored BCE elements will get bypassed and probably, in the end, a GPC will force FTS after logging a non-universal I/O error.

12.5 I/O ERROR CONDITIONS

If the problem involves a total loss of communication, with several BCE elements potentially effected, the scenario will be different depending on the I/O transaction type. In many cases, an initial timeout or SEV bit

error will be logged, and an I/O ERROR message will be annunciated. Usually, if a BTU is powered/failed off, an I/O ERROR XXX will be annunciated. (The MMs are a DPS exception to this rule. An OFF/BUSY MMX will be annunciated at the next MM request if MMX is powered off.) The appropriate I/O RESET command can be also used for I/O error message correction.

The key points to remember about I/O ERROR messages is that they indicate a total loss of input communication with a BTU after BCE initial (first transaction element on the bus) timeout or SEV errors are logged and that all BCE elements for that BTU are bypassed. (The MMs are again notably different in their error signature.) Once again, outputs are normally not affected by input transaction problems.

For ICC I/O errors, the failure of a retry of a transaction will cause a GPC to fail out of the CS, i.e., a CS FTS occurs. There is no I/O ERROR ICCX, but an ICC Retry GPC error code to log the error should it occur.

For DK bus I/O ERROR CRTX messages, polling may continue successfully if the problem is a transient one, or POLL FAIL and BIG X on the CRT may indicate a hard failure. BFC CRT select discrete A or B failures can also result in these indications.

The PL bus I/O ERROR annunciations are based on two consecutive I/O errors being logged. SM has mission-unique I/O error logic implemented, depending on the manifest of BTU's on the PL buses. The LB has no I/O ERROR message annunciation, just I/O errors logged against the SRB and MCIU transactions.

The IP bus has two I/O ERROR fault messages: one for the downlist process (I/O ERROR D/L) and one for the SM PCMMU OI/PL Data RAM accesses (I/O ERROR PCM). The I/O ERROR D/L message usually points to an IP bus transmitter disabled/MIA busy failure, but it could also be indicative of an IP bus problem. If the transmitter totally fails, the BCE downlist program keeps running with no knowledge of the failure and no I/O ERROR D/L fault message is annunciated. However, if this was the SM GPC, the BCE program performing the PCMMU data reads would be logging initial timeouts resulting in an I/O ERROR PCM fault message. For noninitial timeouts, SM annunciates the appropriate S62 BCE BYP XXX fault message after the second consecutive I/O error on the transaction.

The mass memory has a two part I/O ERROR MM1(2) annunciation philosophy. For software overlay and freeze-dry attempts, a preposition failure plus two I/O errors per MM (a maximum of five errors) is possible for MM single-phase overlays depending on the combination of GPC/MMU failures. However, for the two-phase overlays (G1, G2 and G3) and freeze-dry attempts, there could be up to nine tries performed (hence nine I/O error entries logged) with the possibility of nine I/O ERROR MMX messages annunciated (depending on the nature/combination of GPC/MMU failures) if all nine tries are done. It is highly unlikely that all five or nine fault messages will ever be seen, as the five major cycle (4.8 second) fault message lockout window should preclude some of the mass memory I/O errors from repeat annunciation.

This possibility of up to four I/O errors per MMU (plus the pre-position failure) for G1, G2 or G3, due to their two-phase implementation on the MMU, has a future software fix planned for OI-8C. With this fix, if the first phase is unsuccessfully retrieved after two tries, a toggle to the alternate MMU will be done immediately. Also, if the first phase is successful and the second phase unsuccessful after two tries, a toggle to the alternate MMU will be done to retrieve the second phase only. In both scenarios, access time is reduced.

For other MMU transactions; e.g., SM SPEC roll-ins and TFL/DFLs, a single try is made on the MMU specified on SPEC 1 with one I/O ERROR MMX annunciated. One thing worth noting is that for SPEC roll-ins the SM application software first checks to see if the MM is busy, so that an OFF/BUSY MMX fault message is possible and no MMU I/O error is ever logged. A SPEC roll-in request while the SM-assigned MMU is off or busy is the only known case of a fault message being annunciated with no associated I/O error being logged.

All SM "single hit" fault messages have a somewhat strange signature of a Master Sequence Controller (MSC) timeout, rather than a normal initial or SEV timeout. This is actually a broader signature of certain failures with simplex GPC MM transactions in a CS of two or more. For these cases, a MM initial timeout is annunciated as an MSC timeout. The failure must be one where the MM never goes busy, i.e., a GPC transmitter failure, MM bus failure or MM receiver failure.

In FC MDM HFE (25 Hz) input transactions, there has been a return word element added to the front of the BCE element chain. This common element was implemented to detect a total loss of input communication with an FC MDM as fast as possible. If two consecutive return word initial timeouts occur on an FC MDM HFE transaction (40 milliseconds apart), all the BCE elements in all input transactions destined for that FC MDM are bypassed immediately with only an I/O ERROR XXX annunciated. This is done to preclude the fault summary SPEC from being flooded with messages and to prevent erroneous inputs from polluting PASS software and vehicle systems, and it is performed via self-modifying code in the BCE program. Here again, nonuniversal I/O errors logged by a single GPC will cause that GPC to force FTS. PL (and FLEX) MDM transactions also employ the return word feature, but at a rate of 1.04 Hz.

12.6 I/O RESET

The BCE elements of input transactions on the FC strings, payload buses, and launch data buses will be upmoded a bus pair at a time once each major cycle (960 ms). This is provided that these data buses are not bus masked within the major function that that I/O RESET EXEC has been issued. For example, in the case of all in-flight GNC OPS, only the FC strings will be upmoded upon the issuance of an GNC I/O RESET EXEC.

The data bus upmoding will generally be initiated the next minor cycle following the processing of the DEU poll response containing the I/O RESET

EXEC message. This MCDS processing normally occurs within the same minor cycle as the DEU poll, which is during minor cycle 3 or 15. Furthermore, because of the redundant set DK listen capability, the DPS bus upmoding that occurs via the I/O RESET EXEC function does not depend on ICC traffic.

When a FC string or data bus pair is upmoded, the following actions occur during the DPS reconfiguration process. First, all the BCE elements on both of the data buses are upmoded. Next, a mask of "permanent" BCE bypasses is logically OR'ed with the elements of each BCE program associated with the data bus pair. The corresponding BCE programs are then dynamically updated, leaving only those BCE elements that are supposed to be part of the I/O profile for the active OPS. For example, EIUs are "permanently" bypassed in G2, G8 and G3. Finally, the I/O transaction error counters for all 80 potential I/O transactions (from a systems software point-of-view) are cleared, i.e., are reset to zero.

As has been noted previously, SM application software also recognizes a separate I/O RESET to the PCMMU via crew entry or uplink. This was implemented to allow the PCMMU BCE elements for the OI MDMs, PDI and Spacelab to be upmoded following PCMMU reconfiguration. This upmoding is done without potentially triggering nuisance alarms to the crew from the re-bypass of non-IP BCE elements.

12.7 MDM OUTPUT CONDITIONS

MDM output fault detection logic has been implemented for the FC and PL MDMs to provide some protection that the GPC output data path is healthy. This software logic that results in the MDM OUTPUT XXX fault message runs cyclically at 1.04 Hz, which is comparatively low rate when considering the 25Hz FC MDM return word logic. The MDM OUTPUT logic does not run in G9.

A GPC sends a "return word" to the MDMs its commanding imbedded in the other outputs its sending to these MDMs. After an MDM receives the return word, it sends it back to the GPC as a confirmation that the MDM received the commands. In the MDM OUTPUT software logic, the received MDM return word is compared bit-by-bit with the wrap word pattern the GPC put in the return word it transmitted to the MDM. Any miscompare yields the MDM OUTPUT fault message. The wrap pattern could be altered by hardware failures in either the MDM or IOP. The 25-Hz wrap pattern has two bit configurations used alternatively.

The only action the error detection logic takes is to annunciate the failure condition, so nothing is automatically done by this software to preclude the possibility of erroneous GPC outputs, divergent GNC processing and GPC FTSs. However, separate crew procedures delineate the actions taken after an MDM OUTPUT fault message annunciation.

12.8 FCOS I/O ERRORS

When a PASS GPC is moded to STBY, a software module (ARA_SWITCH_MONITOR) detects the discrete and initiates a series of cleanup activities. To bring the GPC down in an orderly fashion (in other words, "nicely"), applications routines are canceled at their next completion. This may take several minor cycles to accomplish. To avoid common set ICC pollution from the GPC going to STBY, its ICC transmitter is immediately disabled.

Depending on the timing involved in disabling this ICC transmitter, a fail-to-sync due to ICC errors may be observed. These will be logged as transmitter disabled in the GPC going to STBY (BCE: ICC bus #, ID: 3 (SSIP ICC), STAT REG; 00000100 (transmitter disabled)) and as MSC timeouts in the other GPCs (BCE: 28 (MSC T/O), ID: 0, STAT REG: STBY GPC's ICC bus bit set). There may be zero to two of these SSIP ICC I/O errors logged depending on the timing. In addition, the GPC going to STBY will likely log a Level A Interrupt I/O error (BCE: 30 (Lvl A), ID: 0, STAT REG: 40000000 (IOP fail latch)) as the GPC sets its I-fail CAM light.

When the GPC is again moded to RUN, a new Level A Interrupt I/O error is logged (BCE: 30 Lvl A), ID: o, STAT REG: 20000000 (IOP control monitor idle)), which is due to issuing an IOP master reset from the FCOS normal initialization routine. If the GPC was moded RUN-STBY-RUN, the time that appears in this error log entry will only be milliseconds more than the time the GPC was moded to STBY. If the GPC was ever moded to HALT, the time that appears in this error log entry will be milliseconds more than zero. The latter is because coming up from HALT, the FCOS initialization routine zeros the software clock.

12.9 RECENT STUDIES

The most recent developments in the area of I/O have concerned RS GPCs and have dealt with improperly terminated (open-ended or shorted) data buses, the loquacious (blabbermouth) BTU and FC MDM A/D Converter logic. The first two cases have been termed diabolical data path failures and can lead to unpredictable non-universal I/O errors with signatures the same as other failure modes, so no special procedures have been put in the FDF to accommodate them.

An improperly terminated data bus causes reflections on the data bus, which can lead to a mixed perception of bus traffic by the GPCs due to their physical placement on the bus. Analysis performed by the Charles Stark Draper Laboratory on Non-Universal I/O Error (Category V) tests that were run at JSC's Software Avionics and Integration Laboratory (SAIL) facility concluded the following on unterminated data buses. An unterminated bus fault has never occurred in flight and, if it did, it would not necessarily cause sync set breakup, sync fail, or even I/O errors. There is no single, unambiguous error signature for unterminated data buses, but there are signatures that qualify as "suspicious." Examples of this are (1) a bypass caused by error seen by fewer than all GPCs and (2) simultaneous universal and nonuniversal errors on both buses of a string. Draper also recommends,

if I/O errors put a bus out of active use and an unterminated fault is suspected, not bringing the bus back into active use. Additionally, they recommend avoiding an I/O RESET in the current OPS whenever possible and avoid assigning the suspect bus in future OPS.

The loquacious BTU, notably an MDM for DPS concerns, causes excess traffic on the bus which interferes with the normal bus transactions. The results of these last two scenarios can be anything from I/O errors and bypasses to GPC FTSs.

FC A/D Converter logic has been implemented to allow the detection of an MDM A/D converter failure. Two analog cards on all of the FF and FA MDMs are sampled to detect the failure. It is annunciated by either BCE STRG X A and B fault messages if an FF MDM A/D converter failure or BCE STRG X C and D fault messages if an FA MDM A/D converter failure. X is 1 through 4 depending on the FC MDM which has the A/D problem. Note that for the OI-8B and OI-8C, because of the way the A/D converter failure software was implemented in the PASS, the BFS will downmode the entire string to which the MDM belongs. A software fix scheduled for OI-8D will cause the BFS to bypass only the PROM sequences; i.e., the PASS and BFS response to the A/D converter failure will be the same. A port mode should fix the problem, as each MDM port/SCU utilizes a different A/D converter. The MDM Systems Brief, section 7.10.1, contains more on this A/D converter failure detection mechanism.

12.10 REFERENCES

1. Space Shuttle Programs Orbiter Avionics Software Operational (DBS) Detailed Design Specifications (DDS), Volume:II - System Services
   Part: 1 - Flight Computer Operating System (FCDS)
   2 - User Interface (UI)
   3 - System Control (SC)

2. National Space Transportation System Computer Program Development Specification (CPDS), HSTS System Level A Software Requirement

3 SYSTEMS MANAGEMENT

SECTION 13
PASS SYSTEMS MANAGEMENT

CONTENTS

Section		Page
13.1	INTRODUCTION	13-1
13.1.1	Major Element Overview	13-1
13.1.2	Concurrent OPS	13-4
13.2	STARTUP, INITIALIZATION, AND CLEANUP	13-4
13.3	PASS SM I/O	13-6
13.3.1	PCMMU Operation	13-6
13.3.1.1	PASS SM Data Acquisition	13-9
13.3.1.2	PCMMU DART Generation	13-12
13.3.1.3	Unique PCMMU I/O Error Logging and Bypass Words	13-13
13.3.1.4	PCMMU Telemetry Format Loads (TFLs), BITE, Read, and Associated I/O	13-14
13.3.2	Payload Data Interleaver (PDI)	13-17
13.3.3	Payload Bus I/O	13-18
13.3.3.1	Low Rate Inputs (1.04 Hz)	13-18
13.3.3.2	High Rate Inputs (6.25 Hz)	13-20
13.3.3.3	PL bus - Outputs to BTUs	13-20
13.3.3.4	PL MDMs	13-22
13.3.3.5	Flex MDM	13-23
13.3.3.6	Sequence Control Assembly (SCA)	13-23
13.3.3.7	Dual port BTUs	13-24
13.3.4	PSP I/O	13-24
13.3.5	MCIU I/O	13-26
13.3.6	SM Checkpoint	13-27
13.4	FDA	13-30
13.4.1	Orbiter Annunciation Overview	13-30
13.4.2	FDA Data Formats	13-31
13.4.3	FDA Data Structure	13-33
13.4.4	FDA Processing	13-36
13.5	PRECONDITIONING	13-38
13.5.1	Timing Concerns with FDA and Preconditioning	13-39
13.6	SCALING AND CALIBRATION	13-40
13.6.1	Transducer	13-40
13.6.2	Signal Conditioner	13-40
13.6.3	Measurement Range	13-40
13.6.4	Measurement Accuracy	13-41
13.6.5	Measurement Digitization	13-41
13.6.6	Cal Curve	13-44
13.6.7	Cal Curves in Flight Software	13-44
13.6.8	Why Cal Curves Change	13-45
13.6.9	Offscale High and Offscale Low	13-46
13.7	SPECIAL PROCESSES	13-47
13.7.1	Payload Bay Doors (SSB)	13-49
13.7.2	APU Fuel Quantity Computation (SSA)	13-50

Section Page

13.7.3 Fuel Cell Computation (SSF) 13-50
13.7.4 Fuel Cell Purge Sequence (SSC) 13-50
13.7.5 O_2, N_2 Quantity Computation (SSN) 13-50
13.7.6 Hydraulic Fluid Temp Control Circulation Pump Sequence (SST) 13-51
13.7.7 Recorder Tape Position (SSR) 13-51
13.7.8 Hydraulic Water Boiler Quantity (SSH) 13-52
13.7.9 Standby Water Coolant Loop Temperature Control Sequence (SSS) 13-52
13.7.10 Antenna Management (SSM) 13-52
13.7.11 Special Processes Data Out (SSO) 13-52
13.7.12 Time Execute Command (TEC) Buffer Poll (PMQ) 13-52
13.8 MISSION SELECTABLE PROCESSES 13-53
13.8.1 SSUS (PAM) Sequence Control (PDS) 13-53
13.8.2 IUS GNC Transfer (PDL) 13-54
13.8.3 Standard GNC Data Transfer (PMG) 13-54
13.8.4 Spacelab Serial Data Interface (SLSIO) 13-54
13.9 REFERENCES 13-58

SECTION 13
PASS SYSTEMS MANAGEMENT

13.1 INTRODUCTION

The development of PASS Orbiter flight software resulted in the partitioning of the in-flight application processing into two major elements: GNC and SM. GNC was tasked with the flight-critical tasks of guidance, navigation, and flight control, and was given the redundant flight-critical strings and associated I/O interfaces. SM was tasked with "performance monitoring" not already included in GNC and payload support; additionally, SM was allocated the Payload (PL) data buses, some MDMs, and an interface with the operational instrumentation system. A number of control processes (PL bay doors, fuel cell purge, etc.) were placed in SM as the most logical place of residence. Since the SM was required to provide PL support, certain SM processing was constrained to be easily mission-reconfigurable with a table-driven approach. These tables are initialized by a ground resident program called the SM preprocessor. The BFS, which backs up the PASS GNC during ascent and entry, also provides a reduced SM function for these dynamic phases. This brief will discuss only PASS SM.

13.1.1 Major Element Overview

This brief is intended to complement the documentation system composed of IBM Detailed Design Spec (DDS) and the level B Computer Program Development Specification (CPDS). In general, "processes" (program routines) which are straightforward and self-explanatory in the DDS will be identified here but not discussed in great detail. Other processes (such as I/O) will be discussed with more detail. In particular, the brief will identify the effect of the SM preprocessor on each element of processing.

For purposes of this brief, SM will be divided into the following major elements:

A. Startup, initialization, and cleanup

B. PASS SM I/O

C. Fault detection and annunciation

D. Preconditioning and scaling

E. Special processes

F. Mission selectable processes

Figure 13-1 shows the SM Control Overview -- essentially, this shows the pathways that SM follows during its activity. Figure 13-2 delineates specific processes within SM as well as the frequency of the processes' execution.

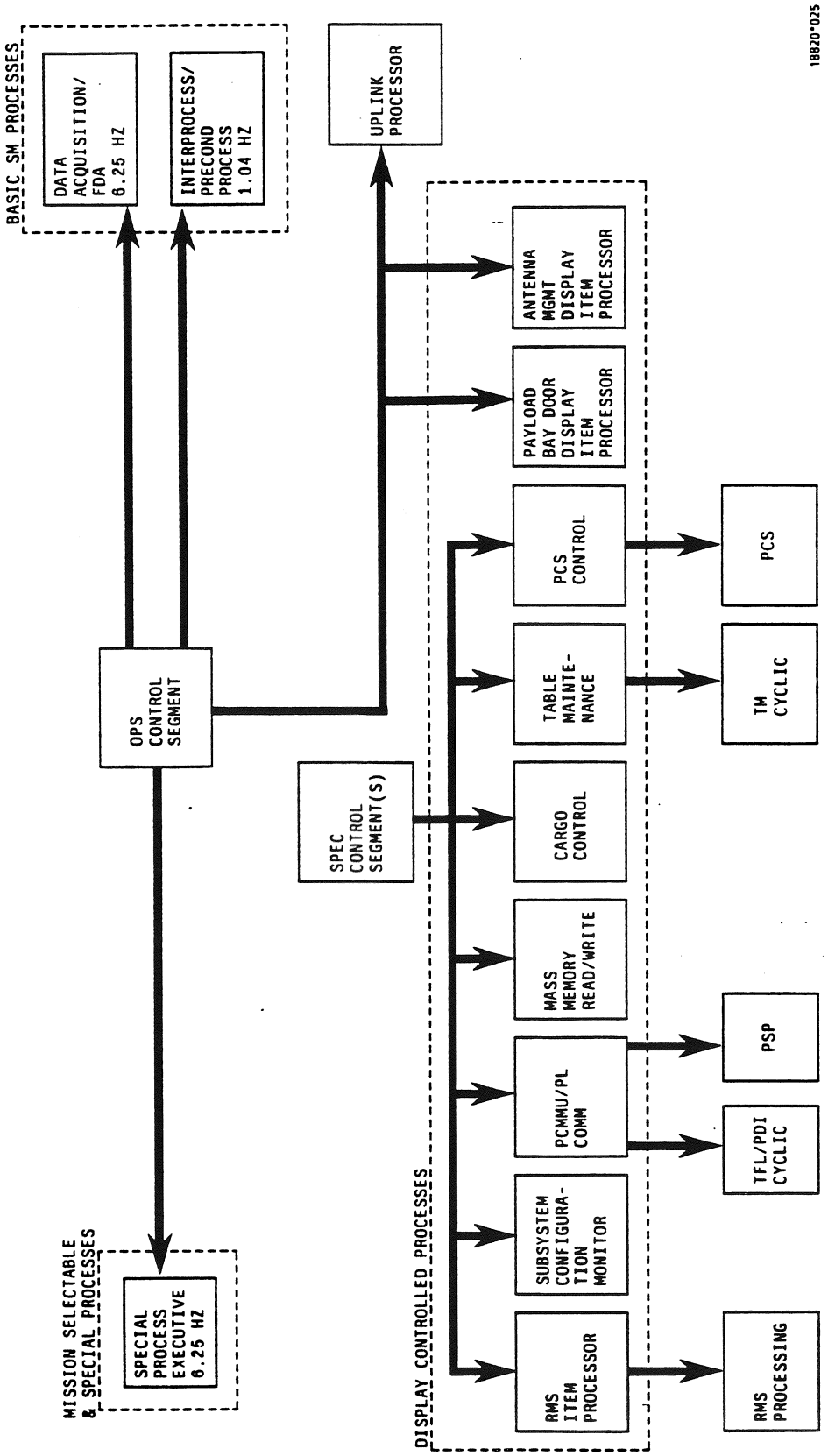


Figure 13-1.- SM Control Overview6.

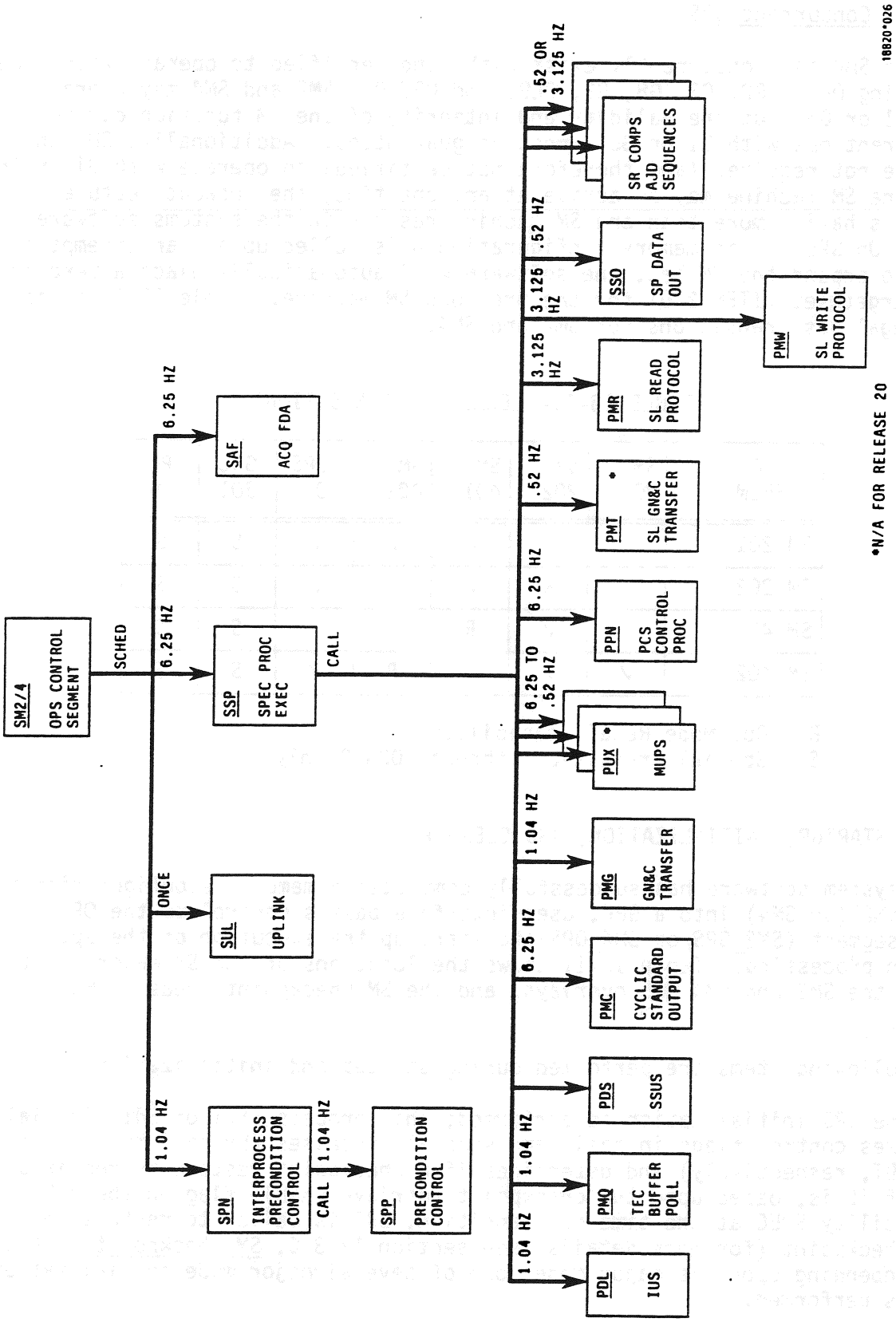


Figure 13-2.- SM Processes6.

13.1.2 Concurrent OPS

SM2 and SM4 can concurrently exist with (and certified to operate with) the following OPS: G2, G3, G8, G9, PL9, and OPS 0. SM2 and SM4 may operate with G1 or G6, but the validity and integrity of the SM function during concurrent ops with G1 or G6 cannot be guaranteed. Additionally, SM2 and SM4 are not required (and therefore not certified) to operate with G1 or G6. Only one SM machine may be active at any one time; the lockout feature that prevents having more than one SM machine resides in the systems software area. On SPEC 0, if memory configuration 4 is called up and an attempt is made to expand the SM set, the software will automatically place a zero in the target set (ITEM 2-6) for the previous SM machine. Table 13-I exhibits the legal ops transitions for SM2 and SM4.

TABLE 13-I.- LEGAL OPS TRANSITIONS

TO FROM	SM 201	SM 202	SM 401	SM 402	OPS 0	GNC 301	PL 9
SM 201	R	✓	✓	✓	✓	S	S
SM 202	✓	R	✓	✓	✓	S	S
SM 401	✓	✓	R	✓	✓	S	S
SM 402	✓	✓	✓	R	✓	S	S

R - Ops Mode Recall capability
S - Special transition; through OPS 0 only

13.2 STARTUP, INITIALIZATION, AND CLEANUP6

When system software has successfully completed a memory reconfiguration to load SM2 (or SM4) into a GPC, user interface passes control to the OPS control segment (SM2_OPS or SM4_OPS) to start up the execution of the application processing. Table 13-II shows the locations of the SM major function base, the SM2 and SM4 OPS overlays, and the SM Checkpoint areas on Mass Memory.

The following items are performed during startup and initialization:

A. The OPS initialization is performed; this process (S2I or S4I) initializes control flags in basic and special processes (by call to SBI and SDT, respectively) and determines if a checkpoint restore is required. If it is, based upon the checkpoint retrieve enable flag on the DPS Utility SPEC at the time of transition, SRE is called to restore the checkpoint (for more details, see section 13.3.6, SM Checkpoint). Also, depending upon the major mode, one of several major mode initializations is performed.

TABLE 13-II.- SM S/W LOCATIONS ON THE MASS MEMORY*

Software type	File	Track	Sub file	Start block	No. of blocks
SM Maj Func Base - Copy 1	3	3	5	0	96
SM Maj Func Base - Copy 2	2	3	5	0	96
SM Maj Func Base - Copy 3	1	3	5	0	96
SM2 OPS Overlay - Copy 1	3	4	0	0	256
SM2 OPS Overlay - Copy 2	2	4	0	0	256
SM2 OPS Overlay - Copy 3	1	4	0	0	256
SM4 OPS Overlay - Copy 1	3	5	0	0	256
SM4 OPS Overlay - Copy 2	2	5	0	0	256
SM4 OPS Overlay - Copy 3	1	5	0	0	256
SM Checkpoint - Copy 1	3	2	0	0	32
SM Checkpoint - Copy 2	2	2	0	0	32
SM Checkpoint - Copy 3	1	2	0	0	32

* This data can be found in the flight specific and OI-release MMU Program Release Notice (PRN).

B. The OPS control segment uses a MACRO to kick off the timer-initiated I/O which is used by SAF (data acquisition and Fault Detection and Annunciation (FDA)). The majority of I/O is initiated on demand in the subsequent application processing.

C. The OPS control segment schedules the major SM element executives:

1. SAF (basic data acquisition and fault detection and annunciation) - see section 13.4

2. SPN (precondition/interprocess control) - see section 13.5

3. SSE (special processes executive) - see section 13.7

When exiting a major mode or the SM OPS, one of a number of cleanups is performed under the control of S2I or S4I.

If the PL and Orbiter support requirements exceed the capacity of a single computer memory, the PASS SM design has several hooks to allow the use of two SM memory configurations. The requirement, as it exists, has two memory configurations, SM2 and SM4, each composed of common processing (Orbiter FDA and displays, etc.) and OPS unique processing. This design permits the PL reconfigurable compools to be segmented into OPS independent (common) and OPS dependent

(unique) portions, much as GNC works now. The major impact is that there are three checkpoint areas: S2, S4, and common.

Historically, the requirement for two in-flight SM OPS was driven by a concern for memory and CPU margin. In light of this, IBM (Spacecraft Software Division) conducted an SM software scrub. The scrub was so successful that the original concerns were virtually eliminated. The community now views the additional SM OPS as an effective means of dealing with the following: excessive payload-related TLM requirements for SM D/L, and an alternate or late manifesting of payloads. For the first instance, the telemetry requirements may exceed the maximum allowable bandwidth of the SM major function for certain combinations of payloads and high/low data rates. Partitioning the payloads between the two SM OPS permits the SM TLM requirements to be satisfied. For the second case (which is more frequently the situation), the final decision regarding which payload flies on a given mission may not be made until a few weeks prior to a flight. Since the lead times required for software development/verification will generally precede such a decision, the SM2 and SM4 implementations can each be tailored for one of the alternate payload manifests.

13.3 PASS SM I/O

The I/O structure which supports the SM function is very different from the flight-critical system for the following reasons:

A. There is very little online redundancy of data content or data path.

B. The SM interacts with a number of data bus devices not used in the flight-critical system. Some examples include the Pulse-Code Modulation Master Unit (PCMMU), the Payload Data Interleaver (PDI), flex MDMs, and Sequence Control Assemblies (SCAs) (fig. 13-3).

C. Many elements of the I/O are mission dependent (depending on whether or not they are used and what data may be acquired).

D. Much of the I/O is on-demand, not continual.

Although such divisions do not exist in the software design, the next subsections of I/O are divided into six sections: The PCMMU, PDI, PL buses, Payload Signal Processor (PSP), and Manipulator Controller Interface Unit (MCIU).

13.3.1 PCMMU Operation[5]

The PCMMU (device ID 10 on BCE 24) generates the telemetry stream and fetches and buffers data by interfacing with the Operational Instrumentation (OI) MDMs, PDI, and Spacelab. Some of these data are acquired by the SM GPC over the IP data bus. These interfaces are described in detail below.

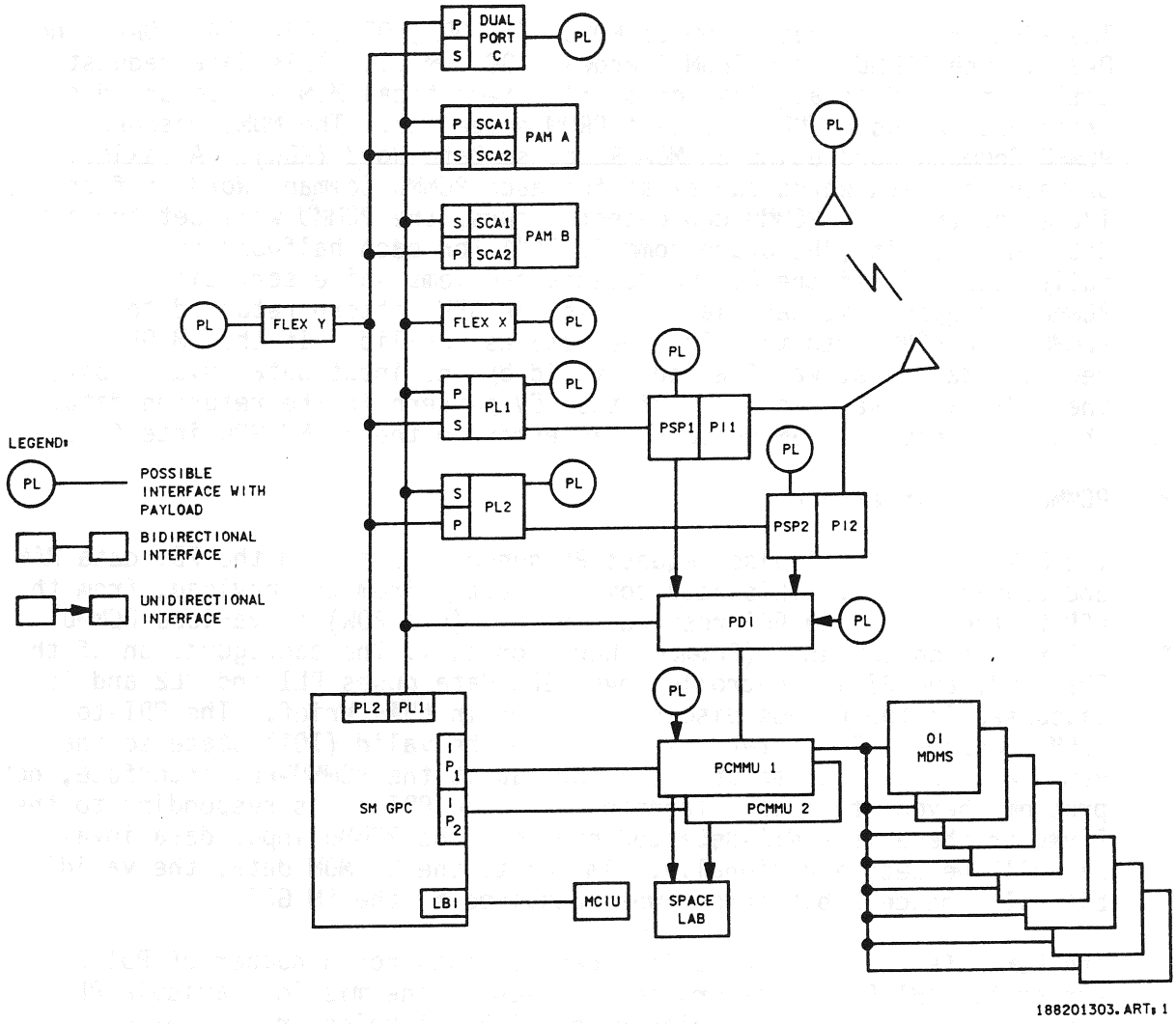


Figure 13-3.- SM I/O Interfaces.

A. PCMMU - OI MDM interface

The PCMMU requests data from OI MDMs OF1, OF2, OF3, OF4, OA1, OA2, and OA3 via the PCMMU fetch PROM (hardware BCE, MSC). This data request is similar to a GPC's acquisition of flight-critical MDM analog and discrete data by using MDM-resident PROM sequences. The MDMs respond to a PCMMU Command Word using an MDM Response Data Word (RDW). A maximum of 32 Response Data Words can exist for each PCMMU Command Word. If an I/O error that the PCMMU can detect occurs, the PCMMU will set the input data invalid bit ("hardware comm fault") for each halfword not successfully acquired. If the OI MDM detects problems while servicing the PCMMU's request, it can flag them in the SEV pattern returned to the PCMMU; the PCMMU can then log the data as invalid. If the SM GPC requests data that was flagged invalid by the input data invalid bit, the PCMMU will set the E bit of the SEV pattern in the returned data. This will induce an SM detected I/O error in the PCMMU-GPC interface.

B. PCMMU - PDI interface

The PCMMU fetch will also request PL support data from the PDI data RAM and toggle buffer. This data comes directly from the payload, from the PSP to the PDI. The PDI responds on call (via RDW) to various PCMMU interrogation commands (Command Word) on call. The configuration of the PDI, PSP, and PI is controlled over GPC data buses PL1 and PL2 and is discussed in the PL bus discussion later in this brief. The PDI-to-PCMMU response SEV pattern is forced to the valid (101) state so the PCMMU will only hear about I/O errors about the PCMMU-PDI interface, not problems beyond the PDI. Therefore, if the PDI stops responding to the PCMMU or there is a MIA-detected problem, the PCMMU input data invalid bit will be set; additionally, similar to the OI MDM data, the validity bit will induce E bit errors when acquired by the SM GPC.

The PCMMU fetch of PDI data includes requests for a number of PDI health and status (BITE) indicators in addition to the mission variable PL data. In order to service high rate (>1 kbps) telemetry of PDI data, the PCMMU telemetry formatter may also access the PDI (through the PDI toggle buffer rather than the PDI RAM) directly. Since this information is not a part of the PCMMU fetch, the data are not stored in the PCMMU data RAM and, therefore, are not available to SM.

A note worth mentioning is that, for most Spacelab flights, the PDI will nominally be powered off. Therefore, the PCMMU fetch will flag the missing PDI BITE data via the input data invalid bit. However, the SM preprocessor is programmed to acquire the PDI BITE status every flight regardless of the PDI being utilized or not. Thus, the E-bit type I/O errors will be detected on the IP bus read transactions, and the element will be bypassed by the SM GPC.

C. PCMMU - Spacelab interface

The PCMMU fetch PROM will include special fetch commands to the Spacelab I/O units' Experiment Computer Operating System (ECOS) and Subsystem Computer Operating System (SCOS) to acquire data originally acquired from the Remote Acquisition Units (RAUs) or generated by Spacelab computers. The same command-response mechanism that the PCMMU uses for the OI MDM and PDI is invoked here as well. The data may be acquired by normal SM acquisition of PCMMU data (similar to the OI MDM method). This data may then be telemetered by the PCMMU telemetry formatter.

D. PCMMU - GPC interface

The PCMMU interfaces with all five onboard GPCs via half-duplex data buses. The MIAs allow simultaneous (i.e., independent of each GPC) asynchronous transfer of data to and from the GPCs. However, the PCMMU will not: (1) allow simultaneous access of the same toggle buffer by multiple GPCs, and (2) simultaneous access of the TLM data format programmable memories (128 and 64 kbps). The same command-response mechanism for other PCMMU interfaces applies to the GPC.

13.3.1.1 SM Data Acquisition (BCE 24, Device ID 10, OP code 9)[1]

SM acquires (at a minimum) several hundred halfwords of PCMMU data (analogs, packed discretes, and some payload digitals) originally from OI MDMs, the PDI, and directly from PLs (i.e., Spacelab). Data acquisition/FDA (HAL module SAF) services FDA and display needs by acquiring 12 unique groups of data (see top row of blocks on fig. 13-4) called read cycles. Six of these read cycles are timer initiated (called TIO) and are distributed evenly over the major cycle (at 160 ms intervals). If the six TIO read cycles cannot perform all the required reads, six read cycles of SVC reads may be used for the overflow, at the discretion of the SM preprocessor. (SAF also includes six read cycles of 1.04 Hz mission-dependent PL bus data.) Each read cycle may acquire up to 24 reads or elements. Each read represents a single request for up to 32 contiguous halfwords from the PCMMU data RAM.

An element will be bypassed by SAF on the second consecutive read error encountered for that read element and is annunciated with "S62 BCE BYP XXX" (table 13-III). Also, if FCOS should encounter initial timeouts on any two consecutive reads, all PCMMU elements are bypassed and "I/O ERROR PCM" is annunciated.

When the SM OPS is invoked, the SM OPS control segment (SM2_OPS or SM4_OPS) schedules the TIO transaction structure and SAF at 6.25 Hz (refer to fig. 13-4). SAF provides pointers to the appropriate starting element of each read cycle entry in the Data Acquisition Read Table (Compool CSA_DART). The DART for OI data reads is a fixed compool which contains one entry per read. This entry identifies a start address within the PCMMU data RAM and the number of additional words to be acquired. To facilitate a mixture of 0.52 Hz and 1.04 Hz, each transaction of the DART contains up to 48 reads.

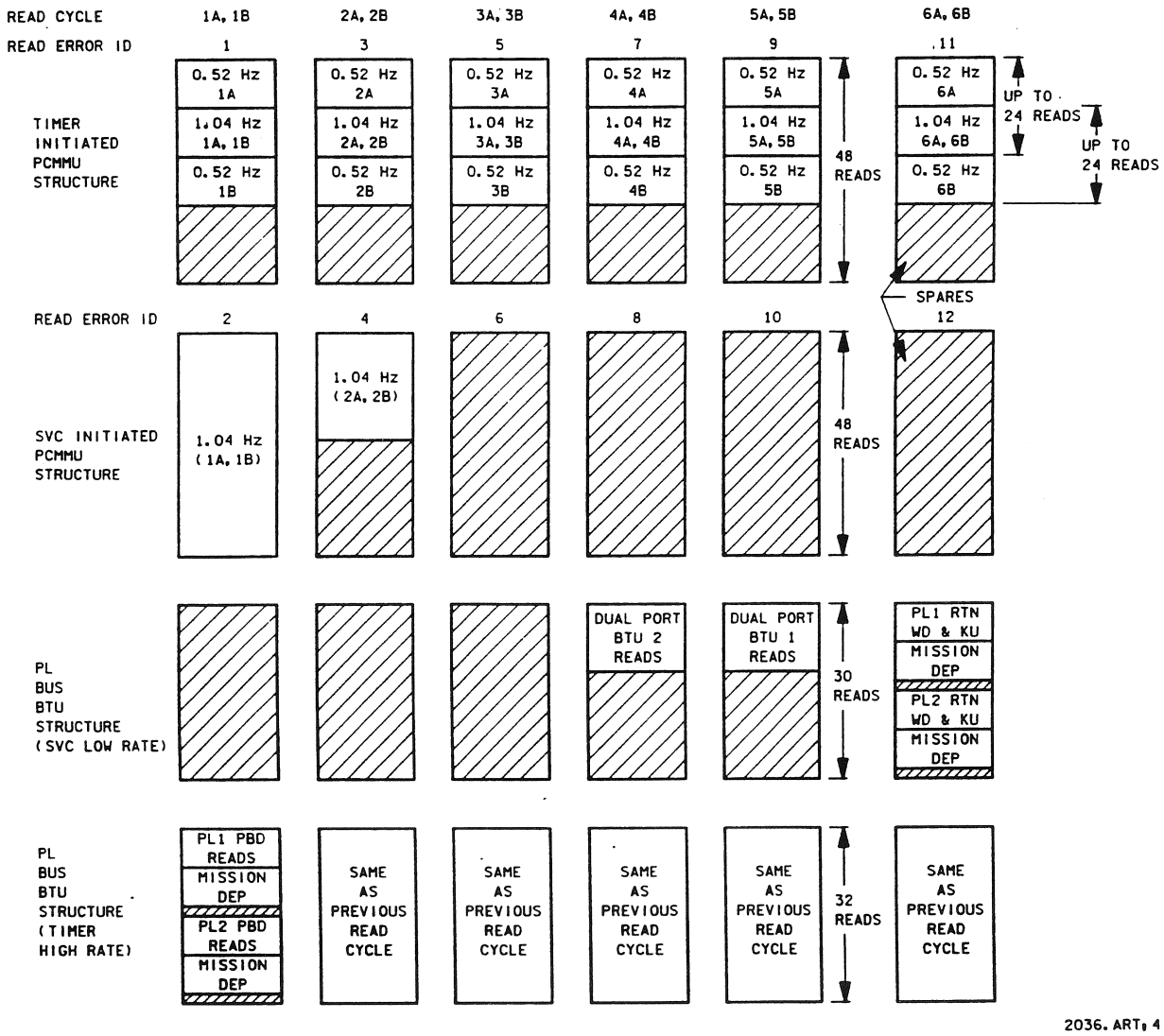


Figure 13-4.- SAF Read Cycle Structure (example)[1].

At I/O completion, FCOS passes the acquired data to SM Compool CSA_INB (input buffer) and indicates which read element (if any) encountered an error. When an error is encountered, that read cycle is ceased for the current major cycle, and any data not acquired thereafter is comm faulted. SVC reads are initiated by SAF, and the subsequent processing is the same as TIO reads. At the completion of each read cycle, SAF moves any data successfully acquired from the CSA_INB to the specific parameter locations in the CSA_PDT (parameter data table) using pointers in a table called the CSA_IXP (input buffer to Parameter Data Table (PDT) cross-reference) and sets up the pointers for the next timer read. SAF will set the comm fault bit in the PDT status word for each parameter not acquired. The PDT structure is discussed more fully in the FDA section later in this brief. If a parameter is to be stored in Engineering Units (EUs), rather than the PCM count MDM format, SAF calls SFS (forward scaling) to perform this computation before storing the value in the PDT.

TABLE 13-III.- OI SYSTEM FAILURE SIGNATURES[5]

Failure or config	Fault message	I/O errors logged	Bypass word indicator
PCMMU or IP fail or PCMMU-OFF	I/O ERROR PCM	2 init T/O on BCE 24	All 12 PCMMU bits
PCMMU OI bus or PCMMU MIA failure	S62 BCE BYP OFA S62 BCE BYP OFB S62 BCE BYP OA S62 BCE BYP D	16 E-bit BCE 24 on BCE 24 (2 for each OI MDM)	All 12 PCMMU bits
OI MDM failure or MDM-off	S62 BCE BYP XXX[1]	2 E-bit on BCE 24	Most of PCMMU bits
OI MDM card failure	S62 BCE BYP XXX[1]	2 E-bit on BCE 24[2]	1 or 2 PCMMU bits
PDI FAIL, or PDI powered OFF	S62 BCE BYP PL S62 BCE BYP PDI	2 E-bit on BCE 24 2 init T/O on BCE 10	Variable number of PCMMU bits; a PDI bit in low rate PL bypass word
PCMMU PL data bus fail	S62 BCE BYP PL	2 E-bit on BCE 24	Variable number of PCMMU bits

[1]xxx = OFA for MDM OF1/2, OFB for MDM OF3/4, OA for MDM OA 1/2/3, D for MDM DC2.

[2]May get several errors if card appears in multiple read elements.

13.3.1.2 PCMMU DART Generation

Since the parameters acquired from the PCMMU may vary and the layout of the PCMMU memory may vary with a PCMMU fetch PROM change, the SM preprocessor must incorporate general rules to build efficient I/O profiles (a DART) according to varying needs. These constraints are as follows:

A. Since SM may acquire PCMMU data at 1.04 Hz and 0.52 Hz, the DART section for a read cycle may be segmented into three sections: 0.52 Hz (section A), an alternate 0.52 Hz (section B), and the 1.04 Hz section. On one major cycle, section A and the 1.04 Hz section will be executed; on the next major cycle, the 1.04 Hz section and section B will be executed. If only 0.52 Hz data are required for SM, the DART for each transaction may contain 48 entries, 24 of which are executed on alternate seconds. If only 1.04 Hz data are acquired, the DART for each transaction may contain only 24 reads which are executed every second. Where the combination of 0.52 Hz and 1.04 Hz data reads exceed the maximum number of TIO entries, the SM preprocessor algorithm will first distribute all of the 0.52 Hz reads across the TIO read cycles described below. The 1.04 Hz reads will then fill the remaining available slots, with the excess of 1.04 Hz data acquisitions placed in the SVC read cycles as described below. Please refer to figure 13-4, SAF Read Cycle Structure, for a graphic illustration of this process.

B. Each read will contain data from only one source: The OI MDMs individually, the PDI, and Spacelab ECOS and SCOS.

C. The preprocessor may generate a read which includes up to three unneeded halfwords (called "holes") in order to combine several short reads into one larger read. The allowed maximum number of holes is selected at the time of preprocessor execution. Inclusion of holes will be used to avoid use of SVC reads.

D. For 1.04 Hz data acquisitions, the following sort/distribution algorithm is utilized:

   1. The read which acquires the most halfwords of data is placed as the first read of the first TIO read cycle.

   2. The next largest read is placed as the first read of the second TIO read cycle.

   3. Continuing this process, the TIO 1.04 Hz read cycles are filled in zigzag fashion: Transaction 1, 2, 3, 4, 5, 6, 6, 5, 4, 3, 2, 1, 1, 2, 3, ... This design generally balances the read cycles by number of reads and number of acquired halfwords.

   4. Reads left over from this process are placed in the SVC read cycles, completely filling each read cycle before starting the next, and starting with read cycle 1 (PL bus SVC reads start with read cycle 6 and work back).

E. For 0.52 Hz data acquisitions, a somewhat different sort/distribution algorithm is utilized:

1. The read which acquires the most half-words of 0.52 Hz data is placed as the first read of the first section A TIO read cycle (i.e., Read Cycle 1A).

2. The next largest 0.52 Hz read is placed as the first read of the first section B TIO read cycle (i.e., alternate Read Cycle 1B).

3. The third largest 0.52 Hz read is placed as the first read of the second section B TIO read cycle (i.e., alternate Read Cycle 2B).

4. The fourth largest 0.52 Hz read is placed as the first read of the second section A TIO read cycle (i.e., Read Cycle 2A).

5. Continuing this process, the TIO read cycles are filled in a square wave fashion: Transaction 1A, 1B, 2B, 2A, 3A, 3B, 4B, 4A, 5A, 5B, 6B, 6A. The cycle is then reversed: 6A, 6B, 5B, 5A, 4A, 4B, 3B, 3A, and so on. This design generally balances the two 0.52 Hz read cycles by number of reads and number of acquired halfwords.

F. The SM/PL Bypass Bit Definition Report is produced every flight, summarizing the DART structure and specifying the mission dependent element numbers (see DPS Shuttle Console Procedure 1.6 on how to look up this data).

13.3.1.3 Unique PCMMU I/O Error Logging and Bypass Words[2]

Since FCOS has no knowledge of the data content of the OI data RAM reads, the FCOS I/O error log makes only the most general statements in the event of read errors of this type. To make up for this limitation, SM provides some special computed downlist parameters to provide data to the ground.

A. ID (V92U0597C) is the read error ID or pointer (1-12) which last encountered an error:

Read Error "ID"

Read Cycle	TIO	SVC
1A, 1B	1	2
2A, 2B	3	4
3A, 3B	5	6
4A, 4B	7	8
5A, 5B	9	10
6A, 6B	11	12

B. EL (V92U0599C) is the element number (1-48) in the read cycle which last encountered an error.

C. TOT (V92U0600C) is a count of the number of SAF PCMMU I/O errors since the last PCMMU I/O reset.

D. PCMMU BYPASS is a word which includes 12 bypass bits, one for each read cycle, which indicates that at least one element in that read cycle is bypassed:

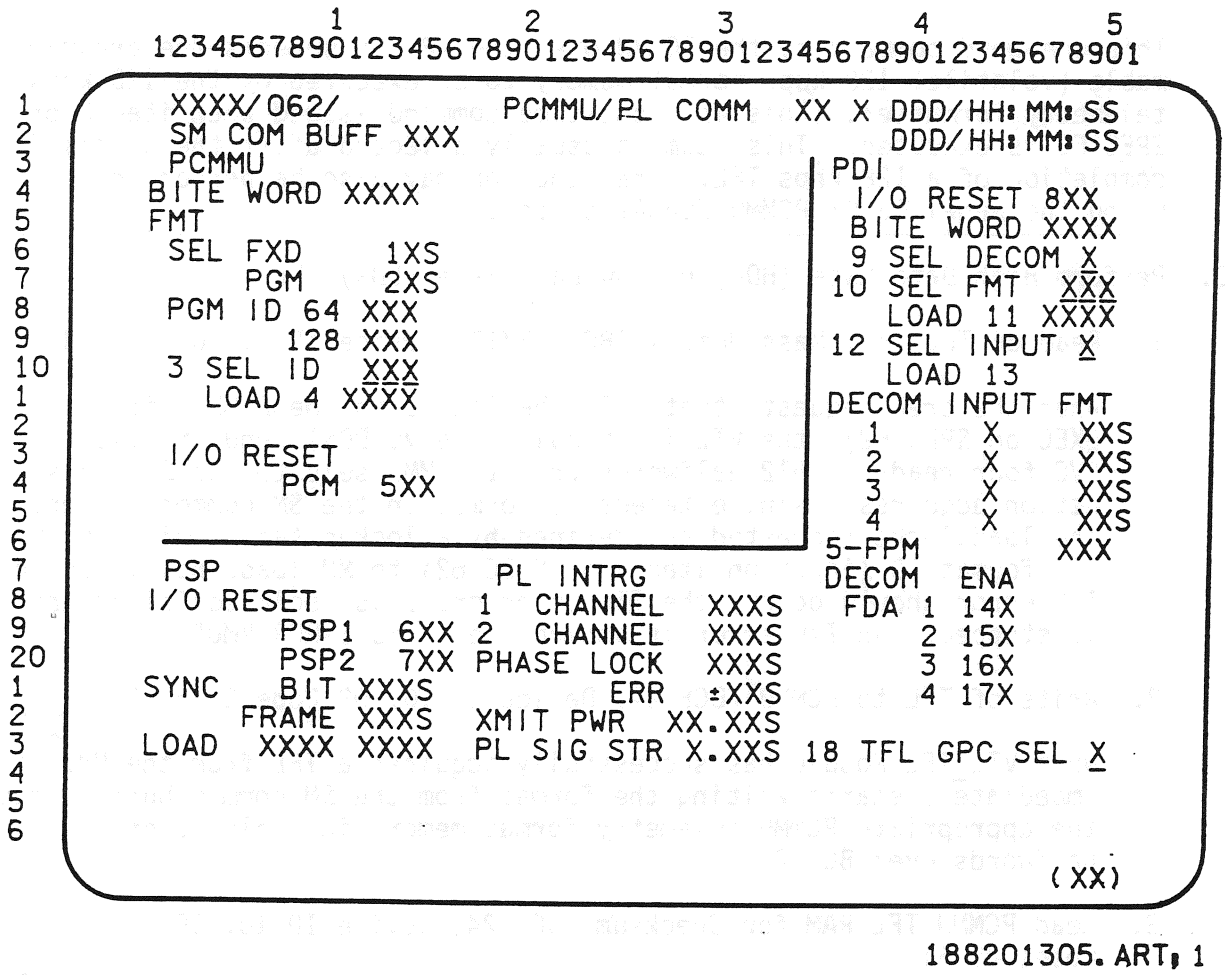
V92X0585X	PCMMU	READ	CYCLE	TIO 1	BYPASS
V92X0586X	PCMMU		CYCLE	SVC 1	BYPASS
V92X0587X	PCMMU		CYCLE	TIO 2	BYPASS
V92X0588X	PCMMU		CYCLE	SVC 2	BYPASS
V92X0589X	PCMMU		CYCLE	TIO 3	BYPASS
V92X0590X	PCMMU		CYCLE	SVC 3	BYPASS
V92X0591X	PCMMU		CYCLE	TIO 4	BYPASS
V92X0592X	PCMMU		CYCLE	SVC 4	BYPASS
V92X0593X	PCMMU		CYCLE	TIO 5	BYPASS
V92X0594X	PCMMU		CYCLE	SVC 5	BYPASS
V92X0595X	PCMMU		CYCLE	TIO 6	BYPASS
V92X0596X	PCMMU		CYCLE	SVC 6	BYPASS

These first 12 bits in parent word V92M0584P indicate the scope of data loss to the SM (see table 13-III). While the SM design includes a bypass bit for each bypassable element, the limited size of the downlist bandwidth prohibits bringing this data to the ground; therefore, these summary words are used instead.

Since the contents of the read cycles vary according to the mission-dependent needs for OI data, and these read cycles are optimized each mission by the SM preprocessor, one needs a mission-dependent document of the structure. The SM/PL Bypass Bit Definition (an SSD IBM deliverable) relates the TIO and SVC ID and EL values to locations within the PCMMU RAM and source device (OI MDM, PDI).

13.3.1.4 PCMMU Telemetry Format Loads (TFLs), BITE, Read, and Associated I/O

Telemetry formats are loaded into the PCMMU's 128 kbps and 64 kbps telemetry format memories on request by item entry to SM SPEC 62, PCMMU/PL COMM (fig. 13-5 (SLS_SPEC)). If no format is currently loaded (these two format memories are volatile), a 128 kbps format (TFL 129 or other fixed formats) which exists in a third nonvolatile memory is used. The telemetry formats are programs stored on the mass memories which the PCMMU telemetry formatters execute to assemble telemetry from OI MDMs, PDI, GPC downlist, and Spacelab.



188201305.ART; 1

Figure 13-5.- SPEC 62 PCMMU/PL COMM[3].

The loading of TFLs involves numerous instances of I/O which are summarized here[1,5]:

A. Select Hard 128 kbps Format (BCE 24, Device ID 10, OP Code 7)

This one-word command to the PCMMU via the IP bus selects the fixed format (PROM) memory to be executed by the 128 kbps formatter. This is a one-time command issued when item 1 of SPEC 62 is executed. For a 128 kbps TFL to be written into the programmable memory, the 128 kbps formatter must be selected by this command or by using the panel C3 OI PCMMU FORMAT switch.

B. Select Program 128 kbps Format (BCE 24, Device ID 10, OP Code 8)

This one-word command to the PCMMU over the IP bus selects the programmable (volatile) 128 kbps format memory to be executed by the 128 kbps telemetry formatter. This is a one-time command issued when item 2 of SPEC 62 is executed. This item is usually selected after the successful completion of a 128 kbps TFL. The function may also be performed by using the panel C3 OI PCMMU FORMAT switch.

C. Perform High Data Rate (HDR) or Low Data Rate (LDR) TFL

1. Read of TFL from Mass Memory (BCE 18/19, Device ID 11, OP Code 3)

When the crew requests that a TFL be loaded to the PCMMU (item 4 EXEC on SPEC 62), the VTL_TFL Module (see VU DDS) requests via the SVC four reads of 512 halfwords from the MMU selected to SM. This action acquires a single telemetry format in the SM common buffer. The loadblocks requested are defined by a lookup table which relates TFL format ID (input on item 3 of SPEC 62) to MM loadblocks. If an I/O error should occur, the TFL in progress is declared failed and is stopped. An I/O error is annunciated "I/O ERROR MMU".

2. Write of TFL to PCMMU (BCE 24, Device ID 10, OP Code 2 and 3)

When VTL_TFL Module has successfully acquired a TFL from the MM, it immediately starts writing the format from the SM common buffer into the appropriate PCMMU telemetry format memory in 4 blocks of 512 halfwords over BCE 24.

3. Read PCMMU TFL RAM for Checksum (BCE 24, Device ID 10, OP Codes 4, 5)

To verify the successful load of telemetry format performed in step 4, the VTL_TFL Module requests a readback with checksum from the PCMMU. If no I/O errors occur and the checksum agrees with the checksum generated on the read from mass memory, the TFL will be declared complete. An I/O error is annunciated "TFL LOAD FAIL" if the checksum is not compatible.

4. Write of Word 88 to 64 kbps Memory (BCE 24, Device ID 10, OP Code 11)

Since 64 kbps telemetry may be downlinked while the format is being changed, the VTL_TFL Module has written zeros into word 88 (the OI sync words) to force the ground telemetry processors to lose lock during the period of write. When the VTL_TFL Module starts the readback (paragraph C.3 above) the module performs a one-time-only write of the real sync word into word 88 to allow the telemetry to be validated by the Mission Control Center (MCC). The PCMMU hardware prevents telemetry format writes from occurring if the program 128 format memory is selected; therefore, no sync word write is required.

D. Read PCMMU BITE (BCE 24, Device ID 10, OP Code 6)

The PCMMU BITE word is an indication of the health of the PCMMU. Since the GPC requires a special BCE program to acquire the BITE word, this acquisition is not included in the SAF acquisition; this step is instead included as a step in cyclic process SPN - precondition/interprocess control. The read is a 1.04 Hz SVC; no other reason exists for including this acquisition in this module except that it happens to run at the desired rate.

13.3.2 Payload Data Interleaver (PDI)[1]

The PDI is used to process payload-generated telemetry streams so that they may be used by SM (by way of the PCMMU). These streams are also buffered for insertion into the operational downlink by the PCMMU. In order to provide flexible interfacing, the PDI offers five input ports to connect with PLs. Each port is selectable to one of four internal decommutators, or decoms. The decom is a simple computer which, when loaded with an appropriate Decom Format Load (DFL), will interpret a particular payload telemetry stream. A particular DFL will cause data to be output to the PDI data RAM for SM, to the PDI toggle buffer for downlink, or to both according to its design. Since the PCMMU fetch of PDI data is fixed from one PDI mission to the next, the mission-dependent PCMMU TFL and mission-dependent SM I/O profile has to access the right data to be compatible with the decom output. To help in this complex interface, a Fetch Pointer Memory (FPM) is loaded with a program which causes data requests from the PCMMU to be serviced appropriately. SM controls the PDI configuration over PL bus 1 (BCE 10), and SM also reads the status of the input switch matrix every 960 ms over this bus. All other PDI status used by SM is acquired from the PCMMU. The PDI has several I/O transactions, some of which are explained below:

A. Cyclic PDI Input Switch Matrix Reads (BCE 10, Device ID 60, OP Code 2)

When SPEC 62 is first activated, the SPEC control segment (SLS_SPEC) schedules VCY_CYC_UPD which reads the PDI input switch matrix for display once every major cycle. Erroring reads are bypassed and are annunicated "S62 BCE BYP PDI". If bypassed, the element may be upmoded by ITEM 8 EXEC on SPEC 62.

B. Perform PDI DFL or FPM Load

1. Read DFL or FPM Load from Mass Memory (BCE 18/19, Device ID 11, OP Code 3)

When the crew requests that a PDI format (item 10 on SPEC 62) be loaded (item 11 EXEC), the VOF_DFL Module (see VU DDS) requests via the SVC the appropriate number of 512 word blocks from the mass memory selected to SM. This modules acquires a single DFL or FPM to load into the SM common buffer. The loadblocks requested are defined by a lookup table which relates load ID (input previously as item 10 on SPEC 62) to mass memory loadblocks. If an I/O error

should occur, "I/O ERROR MMU" is annunciated and the whole process is terminated.

2. Write DFL or FPM Load to PDI (BCE 10, Device ID 60, OP Code 5, 7, 9, 11, or 13)

   When the VDF_DFL Module has successfully acquired a format from the mass memory, this process immediately starts writing the format from the SM common buffer into the decom memory (including the FPM) selected by item 9 on SPEC 62 in blocks of 512 halfwords.

3. Read DFL or FPM Load from PDI (BCE 10, Device ID 60, OP Code 4, 6, 8, 10, or 12)

   To verify successful load of a format load written in the previous step, the VDF_DFL Module requests (via the SVC) a readback with checksum from the PDI. If no I/O errors occur and the checksum agrees with that generated with the read from mass memory, the DFL or FPM load is declared complete. If the checksum miscompares, no fault message will be generated; however, the INCO officer will have telemetry indications of the malfunction.

C. Write of PDI Input Switch Matrix Configuration (BCE 10, Device ID 60, OP Code 3)

   When the crewmember has selected a valid input to decom configuration by SEL DECOM (item 9) and SEL INPUT (item 12), he may implement that configuration by a write to the PDI. This is done by selecting item 13 (LOAD). When this item is selected, the VDF_DFL Module outputs (on a one- time basis) a write of the current state of input to the decom switch matrix configuration (this configuration is monitored cyclically by the previously discussed PDI input switch matrix status read).

D. Although an element exists to read the PDI BITE Status Register (BSR), OP code 1, SM acquires the BITE cyclically through the PCMMU as an element of the SAF read cycle.

13.3.3 Payload Bus I/O[1]

A variety of devices may be connected to PL data buses 1 and 2 (BCEs 10 and 11) for direct SM interface with the PLs and some Orbiter subsystems.

13.3.3.1 Low Rate Inputs (1.04 Hz)

Data may be acquired from the PL bus BTUs at a 1.04 Hz rate by the basic acquisition/FDA module SAF. Like the PCMMU SVC inputs, there are six read cycles of SVC inputs which are built by the preprocessor into DART segments (see SM acquisition of PCMMU data, section 13.3.1.1). Each read cycle may acquire up to 30 reads for a single device ID (15 reads per BTU where 2 BTUs share a device ID for data bus load balancing). The read cycles are filled

TABLE 13-IV.- PAYLOAD BUS ELEMENTS

BTU	Fault message	Device IDs
MDMs PL1 and PL2 (PF1 and PF2)	2	44-52,59
Single Flex MDM	up to 5	56 (Flex 1 and 2 low rate in) 57 (Flex 3 and 4 low rate in) 51 (General high rate in) 59 (General outputs)
SCA pairs and other dual port BTUs including Flex MDM pairs	up to 5 pairs	53 (Dual port BTUs 1 and 2 low rate in) 54 (Dual port BTUs 3 and 4 low rate in) 55 (Dual port BTU 5 low rate in) 51 (General high rate inputs) 59 (General outputs)

by the preprocessor, starting with read cycle 6 (always MDMs PF1 and PF2) and working backwards. For example, if there is also an SCA pair and a flex MDM, read cycles 4 and 5 will contain reads for those 2 sources. Since the PCMMU SVC reads start from cycle one, filling forward, and the PL bus BTUs start from cycle 6, working backwards, the CPU-inefficient SVC reads are more or less evenly distributed over the major cycle. If the device supports MDM return word processing (PF1, PF2, flexes, and flex pairs), the first element in the part of the read cycle for that BTU will be a return word element. If two consecutive return word attempts should encounter errors, all elements for that BTU are bypassed for that I/O transaction and I/O ERROR PL1, PL2, or FLEX is annunciated. If any other element should encounter errors on two consecutive executions, that element will be bypassed and BCE BYP PL1, PL2, FLEX, or SCA will be annunciated as appropriate.

Bypass bits exist in SM for each element. Since there are so many possible elements available to SM and only a few will be active on each mission, the many bypass bits are condensed (logical OR'ed) for downlist only into a 32-bit low rate bypass word (MSIDs V92M0321P and V92M0335P) which includes a return word bypass bit and a single low rate bypass bit for each of the 12 BTUs. The single low rate bypass bit is the "OR" of all individual bypass bits applicable to the BTU on that mission. The element numbers are mission dependent and are defined in the SM/PL Bypass Bit Definition. In addition to these general low rate inputs, there are several specialized input elements which are usually grouped with the low rate inputs, and are included in the low rate bypass words:

PDI cyclic status read (see section 13.3.2)

PSP cyclic status read (see section 13.3.4)

Ku-band serial and fuel cell purge discrete inputs from MDMs PL1 and PL2 (see section 13.3.3.4)

13.3.3.2 High Rate Inputs (6.25 Hz)

If there is a requirement for acquisition of data from the PL bus BTUs at a faster rate than 1.04 Hz, the 6.25 Hz inputs may be used. The transaction is scheduled by the OPS control segment (SM2_OPS) at application initialization to be executed just prior to each SAF read cycle; like the PCMMU inputs and low rate PL bus BTUs, SAF screens the input data. The transaction is composed of 30 individual reads in 2 segments. The two segments are used in slightly different ways depending upon whether the source BTU is dual port and responds to port moding or not. Dual port reads, which may go out either of the two buses (depending on whether the PL buses are selected to primary or secondary ports), appear as the first reads in each segment; in this case, the two segments are device segments. In the case of a single port device such as a flex (which exists on only one bus), its reads must always go out the same bus, regardless of port moding controls. These reads appear at the end of each segment, and the segment may be here viewed as a bus segment. The low-rate read cycles have no need for this system since each I/O transaction or read cycle may contain either dual port data or single port data, but not both.

The SM preprocessor always includes three high-rate elements of PL Bay Door input discretes for each PL MDM on every flight (see section 13.3.3.4).

PL use of high-rate inputs will likely be limited to acquisition for downlist or for a special high-rate mission selectable process.

13.3.3.3 PL Bus - Outputs to BTUs

There is a wide variety of mechanisms for outputs of commands to the PL bus BTUs. The following are the different types:

A. Orbiter subsystem (SM special processes) fixed outputs

These outputs are initiated by SM Special Processes Data Out (SSO) (discussed later in section 13.3.3.4).

B. PSP configuration commands

These outputs utilizing serial I/O channels on MDMs PL1 and 2 are discussed in the PSP section 13.3.4.

C. SCA outputs

These commands are issued to the Sequence Control Assemblies in support of a Payload Assist Module (PAM) deploy. This is done by request of the

SSUS (PAM) deploy sequences (PDS) to standard output control (PMO). The outputs are discussed in the SCA section 13.3.3.6.

D. Standard output control (PMO)

Standard output control is called by any PL process that needs to output serial data through a Standard Serial Interface (SSI). The Payload Control Supervisor (PCS) can also call PMO.

E. Cargo Control (CC) SPEC item entries may be used to initiate outputs over the PL bus BTUs. Each item entry on each CC SPEC has an associated entry in the CPC_GXT (general cross-reference table, a preprocessor generated compool). The GXT entries include a type code which differentiates among the various types of item entries and also includes various indexes to other compools as necessary to implement the request. The types of outputs are as follows:

1. One group of outputs is the analog outputs destined for analog output channels of PL bus BTUs. The engineering unit value input is converted to an MDM compatible PCM count value by call to SBS_BACKWARD_SCALING by the CC SPEC item processor (PCK_ITEM). If the PCM value is within the capability of an analog output channel, the TCS MACRO is invoked to output the value.

2. Discrete outputs initiated by CC SPEC item entry are performed by PCK_CALL to module PCM (CC SPEC discrete item processor). Using the GXT information to index into the CSA (or CS2 or CS4)_DIT to find the set and reset mask, device ID, and channel corresponding to the item number, module PCM invokes the TCS MACRO to output the data. Immediately after sending the set and reset masks, PCM invokes a MACRO to perform BITE Test 4 on all of the discrete output channels controlled in this way. Since a whole set and reset mask is stored for each item entry, any or all bits in the channel may be controlled.

3. PSP outputs may be initiated with CC SPEC item entries (see section 13.3.4-D).

4. Software-pulsed discrete commands are required for certain payloads which latch the state of discrete commands in their own hardware. It is desirable to have the command present for only a brief period. To implement this, there is a variation on discrete outputs implemented by call to PCW. The big difference with regular discrete outputs is that the reset mask is delayed 35 ms after the set mask, and the preprocessor will usually initialize the set and reset command to be the same, yielding a transient output.

5. Hardware-pulsed discrete commands are required to control a special type of output card, available only on flex MDMs, called the Pulsed Output High (POH). These cards only respond to a set command (requested by the PCR - Payload Control Requester) and will reset on their own after a specified interval.

13.3.3.4 PL MDMs

The Orbiter PL MDMs (titled PF1 and PF2 in Master Measurement Data Base (MMDB) documentation) are used to communicate with the PLs using analog and discrete input and output channels that have been set aside for the purpose. These channels are generally labeled Payload Activation, Payload Safing, and Payload Monitor. Serial I/O channels are also available for PL use on cards 8 and 15 of each MDM. The MDMs are used for routine Orbiter functions and have some I/O elements reserved on all flights:

A. Three elements of timer-initiated high-rate PL bay door discrete inputs from each MDM (Device ID 51, elements 85-90).

B. One low-rate return word element for each MDM (Device ID 52, elements 91, 93).

C. One low-rate SVC element of Ku-band antenna status information via serial I/O card 8 of PL1 (Device ID 51, element 92).

D. One low-rate SVC element of fuel cell purge data (the sequence start discrete) from PL2 (Device ID 51, element 94).

   (These four groups of data are included in the I/O profiles by the SM preprocessor every flight.)

E. Some of the standard Orbiter SM Special Processes require outputs (ID 48) through the PL MDMs. These outputs are requested by SSO_SP_DATA_OUT which is called by SSP_EXEC every other major cycle. The discrete outputs are stored and updated by the controlling processes in the CSS_SPOB compool. At the time of output, the stored output card image is inverted and "logical AND'ed" with an applicable bit mask to create the reset mask, with the original stored form representing the set mask. The set and reset masks for each parent word are output to FCOS via SVC request by SSO. This system of applicable bit masks (a modification of the whole word complementing system) allows SM to assure positive control of bits applicable to SM and to leave alone bits not used by SM. A small number of analog outputs are also SVC'ed from this compool to drive Auxiliary Power Unit (APU) fuel and hydraulic H_2O boiler quantity meters with SM-computed values. If a PL bay door sequence or operation is in progress (MM 202), SSO may additionally be called by SSB_PL_BAY_DOORS (the PL Bay Door Special Process) if a PL bay door "STOP" is sensed in the high-rate inputs. This feature is required to ensure a timely stop of door or latch motion if the crew sees something going wrong by having to run SSO at a faster rate than 0.52 Hz.

13.3.3.5 Flex MDM

The flex MDM may be viewed as half of an Orbiter MDM, although there are subtle differences. The flex MDMs essentially consists of a single Sequence Control Unit (SCU) with support for eight Input/Output Modules (IOMs). The flex MDM is typically used to interface with a PL that has many individual discrete and analog inputs and outputs. Flex MDMs are typically located in the PL bay (on a pallet) near the PL. Since the flex MDM has a single MIA on a single PL bus, the MIA is considered a nonredundant interface.

13.3.3.6 Sequence Control Assembly (SCA)[4]

The Sequence Control Assembly (SCA) pair, produced by McDonnell Douglas, is used to control and monitor PAM deploys. A single SCA is a small computer which contains three addressable dedicated serial I/O channels which control various sequences under request by GPC commands.

Each SCA of a pair is wired to one of the PL data buses, and both SCAs have the same IUA (bus address). Output commands are sent out simultaneously on each PL bus to the respective SCAs in the pair; however, only one SCA selected by a hardware switch is actually allowed to output to the Avionics Support Equipment (ASE) which, in turn, controls the deploy of the PAM. The hardware SCA ENA switch also controls the input capability of the SCA pair. If neither SCA has been selected, either unit is capable of responding to an SM GPC poll request. However, when an SCA has been selected, a discrete is sent to the other SCA which produces an interrupt lockout for GPC input requests. During certain SCA status determination operations (i.e., initial powerup, post-deploy temp monitoring), the SCAs are utilized in a non-ENA config. During the PAM deploy operations, one of the SCAs must be selected.

The SM GPC/SCA I/O utilizes a S/W design concept referred to as dual-port I/O. Essentially, the low-rate inputs are requested via only one PL bus at a time; that is, from only one of the SCAs in the pair. However, the software design permits selection of a specific SCA within the pair via PL bus port moding (on the DPS UTILITY spec). The SCA pair I/O interface is similar to the Orbiter MDM in this regard. Since the hardware ENA switch affects the input capability of an SCA, an SCA will not respond to GPC poll requests during deploy operations (unless the unit has been selected). As a result, PL bus port moding and SCA ENA switch selection must be a coordinated activity during PAM deploys.

The SM preprocessor balances PL bus traffic by requesting inputs from SCA1 of some PAM/SCA pairs and from SCA2 of the other PAM/SCA pairs during the same I/O transaction. SCA pairs are considered dual port BTUs in terms of DART entries and bypass words; bypasses are annunciated "BCE BYP SCA".

Outputs are sent to the SCAs in two ways. PDL, the mission selectable SSUS (PAM) deploy sequence, will request at appropriate times the final sequence start and deploy commands by a call to PMO (standard output control). This action outputs the serial data from its residence in a preprocessor generated compool CPA_SCA_CLB (SCA command load buffer) by TCSIO MACRO. Each of

these commands is requested twice by PDL, one major cycle apart, to ensure that these critical commands are effective. Outputs may also be generated by crew item entry. When this particular item is executed on a CC SPEC (ITEM XX EXEC), the SPEC item processor (PCK_ITEM) calls the CC-stored serial command request process (PCL), which, in turn, calls PMO to output data stored in the preprocessor initialized CPA compool.

13.3.3.7 Dual Port BTUs

If a PL requires redundant I/O not supported by the SCA pairs, the generalized capability used for SCA I/O is available for other types of devices. An example would be a pair of flex MDMs. Like the SCA pair, only one of the pair is sent requests for inputs (selected by port moding controls). Unlike SCAs, outputs are not sent to both units; outputs are transmitted only on the PL bus selected by port moding. Since the flex will support return word processing, a return word element is the first read for each BTU. Two consecutive return word errors will cause I/O ERROR FLEX annunciation and will bypass all elements for that BTU. Two consecutive errors on any other element will cause that element to be bypassed and BCE BYP FLEX will be annunciated (dual-port devices other than flexes or SCAs may have characteristics not yet discovered).

13.3.4 PSP I/O[1,4,6]

The PSP, like the Network Signal Processor (NSP), is a two-way serial digital interface device used specifically to output command uplink to a PL. Additionally, the PSP receives PL telemetry from a PL - either by direct hardline to the PL, or RF link to the PL via the PI. The PI is a two-way RF transponder with the following general features: a small antenna at the top of the cabin roof, an approximate range of 20 miles, and no direct interface with DPS. There are two PSP-PI pairs available, only one of which may be selected at a time (like the NSPs). PL telemetry is output from the PSP to the PDI, and PL uplink (as well as PSP configurations) is commanded via a serial I/O channel on Orbiter MDMs PF1 and PF2. There are several aspects of I/O with the PSP:

A. Initiation of PSP I/O

In order to initialize any I/O with the PSP, the crewmember must first perform a PSP I/O RESET on SPEC 62, PCMMU/PL COMM. When this occurs, the SPEC control segment (SLS_SPEC) schedules the PSP cyclic I/O process (SPS_ PSP_CYCLIC) at 12.5 Hz. Normally, only once a major cycle, SPS will schedule a full execution. This includes the cyclic status read (BCE 10/11, device ID 49/50, elements 83/84), although during command load processing, the execution occurs at 12.5 Hz. This read is used to acquire data for crew display and MCC use (through the SM downlist) and is not affected by MCDS keyboard I/O reset, port moding I/O upmode, or OPS upmode until the I/O RESET on SPEC 62 is executed. Once I/O is established, two consecutive I/O errors will cause the annunciation of

an "S62 BCE BYP PSP 1 (2)" and a cessation of attempts to read the PSP. SPEC 62 is also updated to reflect that PSP communication is inhibited.

B. PSP Configuration Outputs (BCE 10/11, Device ID 49/50)

Any time a PSP command load is sent (see paragraph C below), one of five prestored PSP configuration messages is selected to instruct the GPC and PL on relaying commands/data to each other. The PSP configuration specifies data rate, data type, output method (through the PI or umbilical - the PSP has several ports) for command data. This configuration also specifies data rate, frame type and word length, and the sync word for telemetry data. This capability is required to adapt the PSPs as flexible resources to the specific PL needs. The configurations are stored in SM in the process PSP_CMT, which is initialized by the SM preprocessor and may be modified inflight by a unique uplink OP code.

C. PSP Command Load Outputs (BCE 10/11, Device ID 49/50)

Command loads to the PLs may come about in a variety of ways, including PL data load or PL throughput load (uplink OP codes 26 and 61). When a PL data or throughput load is received that specifies PSP as its destination, the data are validated by the SM uplink process (SUL). If the data is found acceptable, the data are written into the PSP common buffer which is a segment of the command load buffer (Compool CSP_CLB). If the PSP I/O has been successfully initiated, SPS_PSP_CYCLIC will output the data (including a configuration message) to the PSP through the PL MDM, thirty-two 16-bit halfwords at a time.

PSP command load outputs may also be initiated by item entry on a CC SPEC. When the CC SPEC item processor (PCK_ITEM) is called to process an item entry, the processor uses that item entry as an index into the General Cross-Reference Table (CPC_GXT). The GXT contains information about what processing is required, and the GXT indexes into other compools for other data. For item entry initiation, the CC PSP command request (PCP_PSP) is called, which in turn calls PSP reserve buffer (SPR_PRB) to apply the indexes from the GXT. The GXT then retrieves the prestored PSP load data out of the Command Load Table (CPC_CLT). Additionally, SM retrieves the corresponding configuration message for the item entry. When the preprocessor builds one of these item numbers, the preprocessor stores the unique command data in the Command Load Table (CLT). If the PSP common buffer is free to implement the output, SPS_PSP_CYCLIC will output the data, as before in paragraph C.

The PCS will also be able to take advantage of this processing in a manner similar to the CC item number.

Interim Upper Stage (IUS) GNC transfer (PDL) and standard GNC data transfer (PMG) (see section 13.8) may initiate outputs to the PSP to transmit Orbiter state vector and attitude information.

13.3.5 MCIU I/O6

The Manipulator Controller Interface Unit (MCIU) is a unique controller/interface device which allows the crew to operate the Remote Manipulator System (RMS) either under SM GPC control or by direct panel control. For SM to control and monitor the RMS, this unique I/O processing must be initiated and maintained.

The cyclic reads for the MCIU (BCE 12, ID 65, element 82) are scheduled by the SPEC 94, PDRS CONTROL, control segment (RMC_SPEC) when the SPEC is first called. The timer-initiated reads occur at 12.5 Hz (every 80 ms) and acquire 45 half-words of RMS status information. The acquisition will continue until the SM OPS is terminated, although FCOS will bypass the element on two consecutive I/O errors. The first time the MCIU is polled by the GPC following a MCIU power-on startup, the MCIU will force the S-bit of the SEV pattern to zero on returned data. This causes a single S-bit I/O error to be logged (no bypass) in the downlisted I/O error log; the S-bit is set on all RDW transmissions. Also following the power-on startup, an echo test word (no. 45) is maintained at zero (a "fail" indication) until a write (BCE 12, ID 66) from the GPC to the MCIU completes. After this sequence of events occurs, MCIU-GPC I/O is established.

Until a crewmember performs an item 5 (I/O ON) on SPEC 94, the input data are not used by the RMS processing, no outputs are performed, and the RMS is said to be in "suspend" mode. When the item 5 is executed, the following events occur:

A. If the MCIU is currently bypassed, the element will be upmoded by SVC to FCOS (an I/O RESET EXEC will perform this although it will not do anything else).

B. The process RQC_IPCD (called by the RMS executive every 80 ms) will take the processing out of suspend mode if the panel configuration is appropriate and will start counting errors (either I/O errors or echo test word failures). If this application error count reaches 2, the RMS processing is forced to suspend mode and the asterisk moves on SPEC 94 from item 5 to item 6. The count is then set to zero if an error-free cycle occurs or if the processing is in suspend mode.

C. If the RMS processing comes out of suspend on one cycle, REX_RMS_EXEC starts performing SVC writes to the MCIU on the next cycle.

While the RMS is used on only a few missions, the RMS processing, including the I/O, is a permanent part of SM2/SM4. When the RMS is not used on a given flight, the MCIU will be removed. Therefore, the processing will always be in suspend, and SM will not suffer the 10-30 percent CPU usage with RMS processing active.

13.3.6 SM Checkpoint[6]

A. Whenever an SM Checkpoint is desired, the Table Maintenance process STM schedules the SM Checkpoint processor SCK. The checkpoint action essentially takes a "snapshot" of various data, and stores the "snapshot" information on the MMUs. In the event of an SM GPC failure, or some other scenario such that SM needs to be moved to another GPC, the checkpointed data becomes an important source for SM reinitialization. The SM checkpoint feature allows the new SM GPC to be configured with the more recent checkpointed data as opposed to using data that was initialized on the MMU preflight. The retrieval of the data is accomplished using the SM Restore function (see section B).

In order to automatically obtain the last checkpointed data if a new SM GPC is required, SPEC 1 ITEM 12 (CKPT RETRV ENA) must be enabled. It is initialized disabled and also disabled after a checkpoint is performed. It is procedurally reenabled after a checkpoint is taken and also if required before the SM software is moved to a new GPC. If this item is disabled and a new SM GPC is needed, the I-loaded data will be brought into this new SM machine. This situation can be remedied by taking the new SM GPC to OPS 0, enabling SPEC 1 ITEM 12, and then PRO'ing back to SM OPS 201. This will bring in the last checkpointed data.

The data to be checkpointed consists of the following:

1. Current FDA status - limit values, noise filter constants, parameter communication enable/inhibit indicators. This data is contained in software compools:

   CSAPDT - SM2/SM4 common (generic) parameters,
   CS2PDT - SM2 unique (flight specific) parameters
   CS4PDT - SM4 unique (flight specific) parameters

2. Table Maintenance (TM) constants - for example, the TDRS state vectors. This data is contained in software compool CSSCDT.

3. Checkpoint timetags

STM reserves the SM common buffer by setting the buffer busy flag if the buffer is not in use; STM proceeds to call SCK. Upon the start of SCK, the Transaction Status Word (TSW) is initialized to prepare for a write execution. Additionally, the checkpoint Read/Write status is set to "R/W in progress", and the checkpoint qualifier to no-op.

The actual checkpoint is accomplished by moving the data from the SM tables to the SM common buffer, and then executing the write to the MMU. Data from the SM tables, located in both the major function base and the OPS overlay, are written to the MMU in separate transactions. The actual process executes the following steps:

1. A Checkpoint is requested via SPEC 60, and the TM display update flag is set. This causes the TM display to reflect the latest checkpoint status.

2. The address of the MMU for the common (MFB) area data is stored in the checkpoint write macro parameter list.
3. The range of MFB FDA parameters to be checkpointed is determined using the Parameter Cross-Reference table (PXT).

4. The internal procedure SCK_COL is called to move the FDA parameters into the common buffer. SCK_COL processes entries in the PXT, and the PXT contains a pointer to the Parameter Data Table (PDT) which has the actual data to be moved to the common buffer. For each entry in the PXT, the corresponding entry in the PDT is examined and one of the following is performed: (1) The discrete data is moved to the common buffer (if a parent, each child is moved), (2) the EU data is moved to the common buffer, or (3) the analog data is moved to the common buffer.

5. The Constant tables are then moved to the common buffer, followed by the mission time.

6. The MMU write macro (CSZV_DATA_WRT) is issued. If the checkpoint is successful (valid checksum), the MFB is updated with the mission time stored in the common buffer. The Checkpoint status remains "R/W in progress", and checkpoint processing then continues. If the checkpoint fails, the checkpoint status parameters are set (Checkpoint R/W status to "FAIL", and checkpoint qualifier to "BOTH"). After the TM display update flag is set and the SM common buffer is freed (by resetting the buffer busy flag), checkpoint processing terminates.

7. The address of the MMU for the OPS-unique area data is stored in the checkpoint write macro parameter list.

8. Using the PXT, the range of the FDA parameters in the OPS-unique area is determined.

9. SCK_COL is called again to move the FDA parameters into the common buffer, and the mission time is then stored in the common buffer.

10. The MMU write macro is then issued. If the checkpoint is successful, the MFB is updated with the mission time from the common buffer and the checkpoint R/W status is set "SUCCESSFUL". If the checkpoint fails, the checkpoint status is set to "FAIL" and the checkpoint qualifier to "UNIQUE".

11. The TM display update flag is set, and the SM common buffer is freed.

An important note that is worth mentioning is that the execution of a checkpoint does not affect the simultaneous processing of TM updates via SPEC 60. However, when values are entered during checkpoint processing, no guarantee exists for the updated value to get checkpointed.

B. If an SM Restore is desired, the SM Control Segments (either SM2 or SM4) calls the SM Restore processor (SRE) to perform either (1) a restore of data if requested via SPEC 1 or (2) a restore of the checkpoint time if the common checkpoint time in the Major Function base = 0.

The SRE process restores the data taken from the last successful checkpoint of the common and OPS unique areas. Using the SM common buffer, the data from both common and OPS unique areas are restored in separate transactions (similar to the data storage for a checkpoint).

Upon the start of SRE, the TRW is initialized to zero for a read execution. SRE then determines whether a restore of SM data is desired, or whether a restore of the checkpoint times is desired. Both actions will

eventually be performed. The actual process is performed by carrying out the following steps:

1. The parameter indicating that the restore was requested by SPEC 1 is turned off. The ICC status is set, and the checkpoint R/W status is set to "R/W in progress".

2. The address of the MMU from which to read the common SM data is stored in the read macro parameter list, and the read macro is issued.

3. If the read fails or if the buffer time is equal to zero (no checkpoint tables), the checkpoint R/W status flag is set to FAIL and the checkpoint qualifier is set to BOTH. The restore process then terminates. If the read fails (Mass Memory read error), a class 3 or class 4 annunciation output is enabled. If the common checkpoint time is equal to zero, a class 5 annunciation output is set.

4. If the read is successful, the common checkpoint time in the buffer is moved to the common checkpoint time in the MFB. The common data is then restored.

5. The range of the PXT is determined for the common SM data, and SRE_DSP is called to disperse the data.

6. The constant tables are moved from the common buffer.

7. If only one OPS-Unique Precondition Table is built for a mission, then the OPS-unique pointer is set and the OPS-Unique Precondition Table is updated.

8. The address of the MMU from which to read the OPS-unique area is stored in the read macro parameter list, and the read macro is issued.

9. If the read fails, the checkpoint R/W status flag is set to FAIL and the checkpoint qualifier is set to "UNIQUE". The restore process then terminates. A class 3 or class 4 annunciation output is enabled.

10. If read is successful and the unique checkpoint time in the common buffer is zero, then no further processing is accomplished. If the time is non-zero, the unique checkpoint time in the buffer is moved to the OPS unique checkpoint time in the MFB and the unique data is restored.

11. The range of the PXT is determined for the OPS-unique area, and SRE_DSP is called to disperse the data.

12. The checkpoint R/W status is set to R/W successful.

With the Checkpoint and Restore algorithms being designed as they are, different checkpoint times for the OPS-unique and common SM are possible; i.e., the common SM checkpoint could be successful and the OPS-unique could fail. However, if an error is annunciated, the checkpoint is usually considered failed for both types of data.

13.4 FDA

The Shuttle program has a goal of as much vehicle "operational independence" as is possible. One feature of this independence is the crew's ability to recognize and recover from failures. Previous space programs featured hardware-based annunciation systems which frequently annunciated normal system operations while generally disregarding the importance of Fault Detection and Annunciation. For these reasons, the Shuttle project allocated DPS resources for a detailed, flexible, and accurate annunciation system that is generally called the performance monitor. The flight-critical performance monitor function is accomplished by GNC (mission critical and PL) Annunciation (GAX). The non-GNC performance monitor is included in the SM function, FDA.

13.4.1 Orbiter Annunciation Overview

In order to standardize indications of failures and to separate more critical events from less critical events, the Orbiter project has specified a hierarchy of annunciations delineated into five classes:

A. Class 1: Emergency

A Class 1 Emergency annunciation indicates a most critical anomaly; this relates to rapid loss of cabin pressure or smoke detected in the cabin or avionics bays. Klaxon alarms are generated due to inputs from the cabin pressure sensors, and siren tones are generated when activated by the smoke detection circuitry. These indications arrive using special coils on the speaker/mike units to ensure that the crew hears them. This annunciation does not result because of an interface with SM software, although the rapid loss of cabin pressure annunciation is backed up by a BFS value computed pressure. This "backing up" is done because the hardware cabin dp/dt sensor is unreliable. Ascent/abort decision-making must be reliable and rapid; rapid cabin pressure loss may influence ascent abort decisions. Therefore, the indications must be reliable.

B. Class 2: Caution and Warning (C&W)

This class is the usual aircraft annunciator panel which indicates, in general, a possible life and vehicle-threatening failure. C&W annunciation takes the following forms:

1. Unique light on the C&W annunciator matrix (PNL F7) by the Caution and Warning Electronics (CWE)

2. Unique C&W tone generated by the CWE which persists until acknowledged

3. The master alarm Pushbutton Indicators (PBIs) (four locations throughout the cabin) illuminated by the CWE

4. The backup C&W light on the F7 matrix driven by the GPC. In order to provide a redundant C&W capability, the Orbiter DPS backs up this function with software which lights the backup C&W light on the F7 matrix, initiates the C&W tone, and generates a CRT fault message. If the measurement is displayed on a CRT display, a down or up arrow is generated in the status field on the display format.

5. While the CWE generates all tones, including emergency and alert, it does have internal redundant tone generators.

C. Class 3: Alert

Since many failures affect only mission success without immediately placing life or vehicle in jeopardy, the Orbiter project-allocated DPS resources for class 3 alerts (incorrectly called SM alert) to detect and annunciate these failures. This alert informs the crew of a situation that may be leading up to a C&W annunciation. The alert routine "pre-conditions" (see section 13.5) parameters with a set of limits, and when the parameter exceeds these limits, an annunciation results. An alert annunciation consists of a fault message; a fixed, finite duration alert tone (generated by the CWE); illumination of the blue alert light on the forward panels which persists until acknowledged; and up and down arrows, as appropriate, on the CRT displays.

D. Class 4: GPC Detected Error

Class 4 generates a fault message only.

E. Class 5: Operator Error (Illegal Entry)

Class 5 generates a message which appears only on a CRT where a keyboard syntax error occurred.

F. Class 0: Not strictly an alarm class since no fault message, tone, or panel light is used; only the CRT status information (up or down arrows) is generated.

FDA can only initiate classes 0, 2, 3, and 4.

13.4.2 FDA Data Formats

Data which are required to be processed by FDA have been placed in the PDT compool by SAF. SAF executes at 6.25 Hz to acquire and, if needed, to scale and/or limit-sense SM data from PCMMU and PL data buses. There are four formats of data so acquired:

A. Analogs

Analog parameters are 16-bit halfwords containing 8 or 10 bits of encoded analog data (generated by an Analog-to-Digital (A/D) converter, usually in an MDM). Ten-bit measurements are in two's complement, and 8-bit measurements are in two's complement, unsigned positive, or signed plus magnitude. These data will have to be scaled (converted to Engineering Units (EUs)) by User Interface some time prior to display on the CRT display.

B. EUs

A 32-bit single precision floating point value which, when directly interpreted to decimal, will yield EUs (°F, psia, etc.). This value is passed to User Interface which may use it directly on the CRT display. Usually, an EU measurement started out as an analog but was converted to EU at the time of acquisition for one of the following reasons:

1. The calibration curve (polynomial which converts from PCM counts to EU) is second or third order (nonlinear) and would present a problem to table maintenance if it were stored in PCM counts. Therefore, conversions are time consuming with a high susceptibility for errors. Since table maintenance updates start with EUs and must interpret them to the storage format, the algorithm (SBS - backward scaling) would have to cope with multiple or complex solutions in factoring a second or third order polynomial. To prevent this problem, these measurements are stored in EUs and are updated directly with no conversion.

2. The cal curve is linear but negative slope. This means that the highest PCM count value is the lowest EU value. If this format were stored in PCM counts, the upper EU value would be computationally treated as a lower limit. Storing these parameters in EUs prevents this algebraic catastrophe.

3. The parameter is required as an EU input to SM Special Processes.

C. Discretes

When transferring data throughout the avionics system, it is most efficient to combine many discretes in a single halfword (rather than use a whole halfword to store a single bit, as IBM frequently does, internal to the GPC). This packing is done at the point of digital origin, e.g., MDM. This philosophy does, however, complicate the subsequent use of the discrete in that the user must get the halfword (called "parent" word) and figure out which bit is of interest. This structure implies that the data must be - at least briefly - doubly stored; that is, as both a parent and individual bit. An 8-bit parent may reside in a 16-bit word along with any one of three types of 8-bit analogs.

D. Digital data

This is just a catchall for unique formats which do not fit the previous categories. Digital parameters consist of 2 to 64 contiguous bits within contiguous 16-bit words.

13.4.3 FDA Data Structure

The key to FDA processing is the intricate table structure, built by the SM preprocessor; the heart of these tables is the PDT. PASS SM has separate tables (e.g., one for upper limits and one for lower limits) while BFS puts all limits in one "super" table. Similar to all potentially OPS dependent SM preprocessor compools, this PDT has two sections; CSA_PDT for the Orbiter parameters and CS2_PDT for payload parameters. If SM4 existed, then there would also be a CS4_PDT for SM4. Figure 3-6 illustrates the various types of entries which correspond to the various data types (digitals take an analog format). Tables 13-V and 13-VI illustrate the structure of FDA information and status words. The SAT pointer is an index into the CSA (or CS2) SAT (scaling table which stores cal curves). This index is used either by SAF to scale EU measurements or by table maintenance for display and backward scaling of limits for analog measurements.

TABLE 13-V.- FDA INFO (ANALOG/EU)*

Bit	Title	Meaning
1-8	PARMID	The most significant 8 bits of the SM parameter ID used for table maintenance. The remaining 16 bits are stored in CSI_PXT (or CS2_PXT) by the preprocessor.
9	HARD FAIL	Set by SAF when the measurement has been out of limits for the noise filter duration and has been annunciated.
10	ALARM INHIBIT	Set by table maintenance INH/ENA item numbers for each measurement.
11-14	NOISE FILTER	Set by preprocessor: The parameter must be out of limits this many consecutive FDA executions to annunciate. (May be changed by table maintenance.)
15-16	ACTIVE LIMIT SET	Set by SPP (preconditioning) to indicate which limit set is to be used.
17-26	FMPT INDEX	Set by preprocessor as the pointer to the fault message required if the measurement is out of limits.
27-32	# OF LIM SETS	Number of limit sets (by preprocessor).
29-32	CURRENT N COUNT	The current count, by SAF, of how many consecutive FDA executions the measurement has been out of limits.

*Discretes include these and others.

TABLE 13-VI.- STATUS WORD

Bit	Title	Meaning
1	BLANKING	Set to tell display format to blank this value (not often used).
2	I/O VALIDITY	Set by SAF to say data were not acquired (comm faulted, display will present "M").
3	OFF SCALE HIGH	Set by SFS to indicate analog measurement is at highest value (display will present "H").
4	OFF SCALE LOW	Set by SFS to indicate analog measurement is at lowest value (display will present "L").
5	OUT OF LIMITS HIGH	Set by SAF if measurement is out of limits high (display will present "↑").
6	OUT OF LIMITS LOW	Set by SAF if measurement is out of limits low (display will present "↓").
7	REFORMAT FLAG	Set by preprocessor to indicate parameter is EU stored.
8	FDA	Set by SPP (preconditioning) to indicate parameter is to be FDA'ed (it is not currently in precondition limit sensor time delay period).
9	REPEAT	Set by preprocessor to indicate a pair of 8-bit analogs are packed in a single halfword.
10	START BIT	Set by preprocessor to identify whether this is the second PDT entry of a pair of packed 8-bit analogs.
11	STEADY STATE (Parent only)	Set by SAF if FDA bits in parent are unchanged (considering noise filter).
12-13	TYPE (Analog/EU only)	Set by preprocessor, indicates input format of analog and scaled EU parameters. Type and corresponding flag values are: - 10-bit unipolar (0 to 500 PCM)...-"001" - 10-bit bipolar (-512 to 511 PCM) -"010" - 8-bit two's complement.....-"011" - 8-bit unsigned.....-"100" - 8-bit sign plus magnitude.....-"101" Analog type does not apply to interprocess EU parameters.
14-15	TYPE	Set by preprocessor. Indicates type of parameter. Type and corresponding value are: - Analog .....-"00" - Discrete parent -"01" - EU .....-"10"
16	UI FLAG	Set by preprocessor to prevent UI from using any bits other than 1-6.

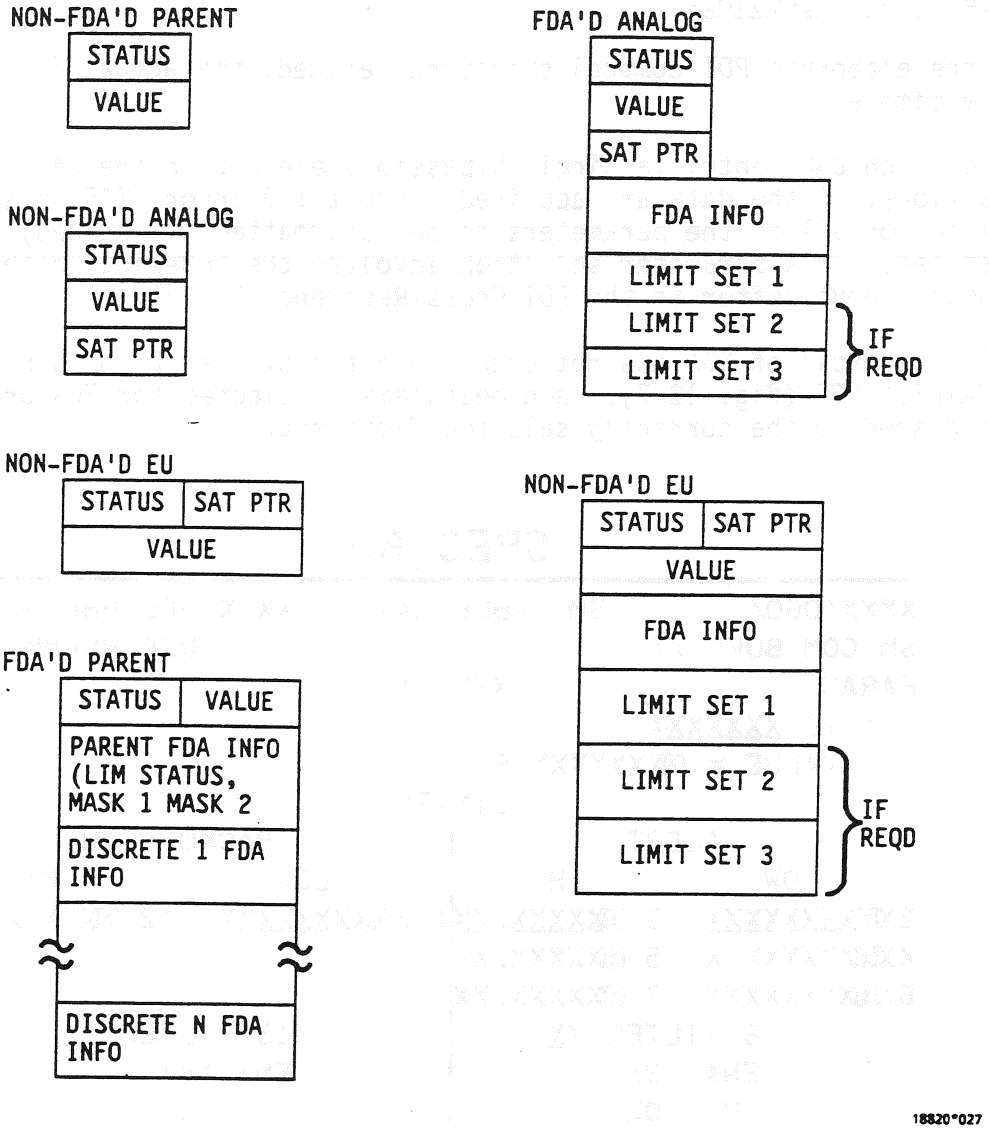


Figure 13-6.- PDT typical entries[1,6].

Parameter Data Table (PDT) COMPOOL: CSA_PDT

The PDT contains an entry for most parameters acquired for processing by Basic SM. This includes the following types of parameters: FDA'ed parameters, precondition inputs (except Interprocess Precondition inputs), displayed parameters, Special Processing inputs, and SCM inputs.

All PDT entries contain a status and a value. Analog and EU entries also contain a Scale Table pointer. The status words contain status indicators and parameter type indicators. Entries for FDA'ed parameters contain FDA information (noise count, annunciation enable/inhibit, etc.) and limit values.

13.4.4 FDA Processing[1]

With the elaborate PDT compool structure defined, the actual FDA process is fairly simple:

A. For each DART entry (a single bypassable element in the SAF data acquisition), if the data are acquired without I/O error, SAF sets up a DO loop for all of the parameters to be reformatted (if needed) and limit sensed. This parameter selection involves the indexing in the IXP compool (input buffer to the PDT Cross-Reference table).

B. If the whole of FDA has not been inhibited by item 20 on SPEC 60, TABLE MAINTENANCE (fig. 13-7), each measurement selected for FDA processing is compared to the currently selected limit set.

SPEC 60

```
XXXX/060/          SM TABLE MAINT   XX X DDD/HH:MM:SS
SM COM BUFF XXX                        DDD/HH:MM:SS
PARAM                    XXXXXX
  1 ID XXXXXXX
    VALUE = XXXXXXXXS
                         LIMITS
          ALERT                          BACKUP C&W
     LOW          HI                LOW          HI
2X XXXXXXXX   3 XXXXX.XX  11X XXXXXXXX   12 XXXXXXXX
4X XXXXXXXX   5 XXXXX.XX
6X XXXXXXXX   7 XXXXX.XX
        8 FILTER XX                    13 FILTER XX
        ENA  9X                        ENA 14X
        INH 10X                        INH 15X
CONSTANT
  16 ID XXXXXXX
  17 VALUE = XXXXXXXXXXXXXXX
CHECKPT  XXX/XX:XX:XX     FDA
  INITIATE 18              ENA 19X
  STATUS XXXX              INH 20X
                                          (XX)
```

188201306.ART: 1

Figure 13-7.- SPEC 60 SM TABLE MAINT[3].

C. If the measurement value is greater than the upper limit or less than the lower limit of the currently selected limit set, the appropriate out-of-limit bit (Offscale high bit and Offscale low bit, respectively) in the PDT status word is set and the current noise count is incremented.

The SM maintains a single "out of limits" (OL) indicator (CSAS_PDTA_HARD_FAIL) for out of limits high and out of limits low. The HARD_FAIL indicator changes only after the "in limits" or OL status has changed for that parameter's particular noise count FDA cycle. Consequently, an FDA message is annunciated only when the HARD_FAIL indicator changes from OFF to ON.

Currently, the FDA performs OL value detection as such: The routine that determines if a value is OL high is always executed first. If the limit is found to be OL high, then the HARD_FAIL indicator is set to OL high. The routine that determines if a value is OL low is not executed if the value is found to be OL high. If the value satisfies the upper limit test, then and only then is the OL low routine processed. The HARD_FAIL indicator would then be set to OL low if the value is found to be OL low after the duration of the noise count.

D. If the current noise count is greater than the noise filter value (controlled by items 8 and 13 of SPEC 60), the HARD_FAIL indicator in the PDT is set and, if the individual parameter annunciation has not been inhibited (by items 10 or 15 of SPEC 60), annunciation is requested. Once the HARD_FAIL indicator is set by an acquisition, the noise count is reinitialized only when the parameter is within limits for the duration of the noise filter.

Some subtle nuances pertaining to the FDA limit processes are worth mentioning. As previously stated, if a value is determined to be OL low, the HARD_FAIL indicator will be set to OL low; consequently, a message will be annunciated. Now if the current value becomes OL high (possibly due to a TMBU) before the noise count has been reset, the HARD_FAIL indicator will change from OL low to OL high. However, no new message annunciation will occur. Similarly, if a value is OL high (with the HARD_FAIL indicator set to OL high) and then become OL low (assuming that the value is no longer OL high), the HARD_FAIL indicator will change from OL high to OL low. If by some bizarre chance that a value is both OL high and OL low, the value will be classified as OL high since the routine to detect a OL high value is executed first. Again no new message will be annunciated until the noise count has had a chance to reset and subsequently set the HARD_FAIL indicator to OFF.

E. To request FDA annunciation, the FDA processing uses the FMPT index from the PDT and a user interface MACRO to set a bit in the CDL_ANNUN compool. At regular intervals, the user interface annunciation processor (DMT_ERR) inspects these bits and generates annunciation as appropriate. Each bit represents a unique combination of major field, minor field, and alarm class.

F. Since the limit testing is always done, the out-of-limits flags in the status words reflect the current moment-to-moment status of the parameter, regardless of the status of the HARD_FAIL indication. Also, if the whole of FDA is inhibited, the limit status is frozen.

G. Discrete measurements packed in a parent word are handled in a slightly different way. A second DO loop is established to examine the FDA'd bits in a given parent. The PDT for discretes includes a mask which identifies the bits in the parent word to be FDA'ed and a set of FDA information for each such bit. Instead of a lower and upper limit, there is only an "undesired state" which is used to examine the current state.

H. SAF uses FAULT_DETECT_ANNUN to process measurements acquired from the PCMMU, or via the PL buses. Measurements created by SM execution (for example, APU fuel quantity values) are gathered and limit sensed by SPN_INT_CONTROL (preconditioning and interprocess control) which uses FAULT_DETECT_ANNUN to FDA these interprocess parameters from various compools. These parameters are located using the Interprocess FDA Table (IFT) which relates the PDT entry to the address where the current value of the parameter is stored.

The PDT and IXP are built by the SM preprocessor based upon SM and PL FSSRs. FDA limits, noise filters, alarm classes, and fault messages are defined here.

13.5 PRECONDITIONING[1,6]

To prevent nuisance annunciation due to normal system reconfiguration (primarily turning a device on or off), the SM FDA provides up to three limit sets for classes 0, 3, and 4 (not class 2). Only one of the three limit sets is actually used at a given time, the one selected by the precondition process. This process is summarized as follows:

A. SPN_INT_CONTROL (preconditioning and interprocess control) is scheduled once per major cycle by the OPS control segment while FDA is enabled.

   If FDA processing is enabled (item 19 of SPEC 60), SPP (precondition processing) is called. Additionally, each interprocess FDA parameter identified in the IFT is FDA'ed.

B. The precondition information is stored in the preprocessor-generated Precondition Control Table (PCT). For each limit set, there exists a logical statement comprised of the AND'ing or OR'ing of up to four parameters - either discretes (which are 1 or 0, true or false) or analogs and EUs (which are in or out of limits). For analogs, this determination is usually derived from the PDT status of class 0 limit sensing of the analog measurements.

C. The three logical statements in a precondition group associated with an FDA parameter are each evaluated. If a measurement used in a statement is invalid due to I/O error; i.e., no statements are true, then the first limit set is selected. If only one statement is true, the corresponding limit set is selected, depending on which statement is correct. If more than one statement in the group is true, the first limit set is selected. Selection of the appropriate limit set occurs by updating the limit set selected field in the FDA information word (bits 15 and 16) discussed earlier.

D. To allow a system some time to respond to a reconfiguration, a warmup time delay is introduced every time the system switches limit sets. During this interval (specified by the preprocessor in the PCT compool and unique to each parameter), no limit set is selected and FDA limit sensing is prevented. If by some chance the configuration should change again during a warmup time delay period, the preconditioning processing will not respond until the end of the delay period. The precondition processing will then enter another delay period. For this example, this means that a transient precondition measurement change can cause loss of the FDA monitoring for twice the limit-sense time-delay period. Of course, this phenomenon can concatenate. Limit-sense time delays of over 30 minutes actually have occurred, so the consequence of a glitching precondition measurement can be severe.

The precondition statements are generated in the SM compools by the SM preprocessor based upon requirements in the SM and payload FSSRs. If a precondition group is used by more than one FDA measurement, the preprocessor recognizes the fact and only builds one copy of the preconditioning design.

13.5.1 Timing Concerns With FDA and Preconditioning

Anytime more than one measurement is used to decide anything, there is always the danger that information about a changing situation will "trickle in" over a period of seconds because the measurements travel through different paths. If the noise filter and limit sense time delays are large enough to account for these varying data flow delays, the only problem should be a delay in annunciation. If, however, the noise filter is small, say 2 or 3, the measurement can show up to FDA long enough before the preconditioning performs a switch to the alternate limit set to cause a nuisance annunciation. The cause of this annunciation will probably not be apparent to the MCC because the telemetry process will tend to obscure the timing. The delays occur due to the following:

A. The SAF-controlled FDA and preconditioning run at different times, as much as a major cycle apart for some parameters.

B. SM FDA and preconditioning operate on a 960 ms major cycle and the PCMMU operates on a 1.00 second cycle. This means that the relationships between these two elements change on a 25-second cycle, and that a nuisance may not be repeatable.

C. The Orbiter PCMMU and the SMS PCMMU model do not exhibit the same timing, so behavior in the SMS and the Orbiter will be different and cannot be used interchangeably.

D. The PDI and the PCMMU store data between the time of input and the time of output, typically a maximum of 1 second.

E. The original sampling of the transducer (either FDA measurement or preconditioning) occurs at various intervals. If the sampling happens immediately after an event, no delay occurs. If the sampling occurs right before an event, the event will not be sensed until the next sample; therefore, a delay may be introduced.

F. The PL and flex MDMs and SCAs exhibit only sampling errors due to reasons outlined in paragraph E above, although some PLs may store or delay the measurements in unique ways.

13.6 SCALING AND CALIBRATION

The use of analog measurements involves many translations from the original form (i.e., the temperature of a chunk of metal), to a usable decimal readout on a CRT screen. Since the SM design is tasked with a part of this job, it is necessary to explain the whole process to see the relationships. The first few paragraphs are some definitions.

13.6.1 Transducer

A measurement is created by immersion of a transducer in the subsystem environment. The transducer converts the subsystem parameter (pressure, temperature, voltage, etc.) into some varying electrical quantity (voltage, resistance, frequency) which is a function of that subsystem parameter.

13.6.2 Signal Conditioner

The electrical quantity generated by the transducer is normalized by a signal conditioner. The signal conditioner turns this quantity into a voltage which is compatible with the A/D converter of the MDM. Most of the analog Orbiter measurements (including flight-critical analogs) rely on the standard OI Dedicated Signal Conditioners (DSCs). Many measurements (all PL measurements) utilize unique signal conditioning equipment which has characteristics unique to that device. Signal conditioners are also used to adjust and buffer measurements for other purposes, in particular panel meters and input to the CWE unit (primary C&W).

13.6.3 Measurement Range

When a measurement is required to be included in the Orbiter, the most important specification is that of range; i.e., the maximum and minimum

values needed to monitor subsystem performance. Many times this selection is a compromise between the widest range expected in the course of missions and a particular range of interest; there are some measurements that spend most of the mission outside of the spec range.

Also, for economic reasons, it is beneficial to use standardized measurements "off the shelf" wherever possible, so some measurements will have range capability that is not used. To achieve the desired range, the signal conditioner may be adjusted in two ways:

A. The transducer range is converted to a -5.00 to +5.00 V range (normal bipolar).

B. The transducer range is converted to a 0.00 to +5.00 V range (most OI measurements are of this type and are called unipolar).

13.6.4 Measurement Accuracy[4]

A secondary specification on analog measurements is the sensor accuracy. Normally, accuracy only deals with the quality and stability of a measurement; hence, if a sensor's characteristics do not change, an exact calibration can be made in the laboratory and the measurement may be considered to have accuracy up to the precision of the reading. However, several problems may occur that prevent this level of accuracy:

A. Some measurements exhibit hysteresis (memory effect); i.e., the function relating the system parameter to the electrical quantity is different for increasing values than for decreasing values.

B. Some measurements sense more than one parameter; i.e., a pressure measurement that is slightly sensitive to temperature.

C. Some measurements change characteristics with age.

D. Quantity and flow rate measurements are inherently unreliable and can change characteristics when least expected.

E. All measurements which are digitized by the MDM A/D converter can be inaccurate to the value of the Least Significant Bit (LSB).

Measurement accuracy is used to warn the user about the expected extent of these variations (usually as a Root Sum Square (RSS) of the errors). By groundrule, the flight rules which define failures in terms of measurement values are required to consider these errors.

Measurement accuracy is also used to allow many individual transducers of the same range and type to be identified with a single performance characteristic called a "standard" calibration curve. Even though each transducer will have a unique characteristic, production quality control can ensure that individual variations fall within the accuracy band. Previous space

programs provided a unique calibration curve for each transducer (see later section on cal curves).

13.6.5 Measurement Digitization[4]

The output of the signal conditioner is connected to one channel of an analog input card of an MDM.

Figure 13-8, although highly simplified, represents the basic analog input signal processing performed by an MDM.

An analog input to an MDM is always a dc voltage, not to exceed 5.12 V dc. The analog input signal interfaces with the MDM via one of two types of analog input cards, Analog Input Differential (AID) or Analog Input Single-Ended (AIS).

A. If to an AID card (16 channel pairs, 0-15):

This card requires a two-wire analog input. Normally, neither input connection will be at analog ground. Thus, analog signals that range positive (+) or negative (-) relative to the analog ground can be detected via this type of IOM card. Examples of such analog signals interfacing with an AID card are aerosurface position feedbacks, rate gyro signals, RHC R, P, Y input commands, etc.

B. If to an AIS card (32 single channels, 0-31):

This card requires a single-wire analog input with the analog ground pin connection serving as the electrical return. Thus, analog signals interfacing with this type of IOM are always referenced to the analog ground. Examples of analog signals interfacing with an AIS card are propellant tank pressures and temperatures, RCS engine chamber pressures and temperatures, OMS actuator position, etc.

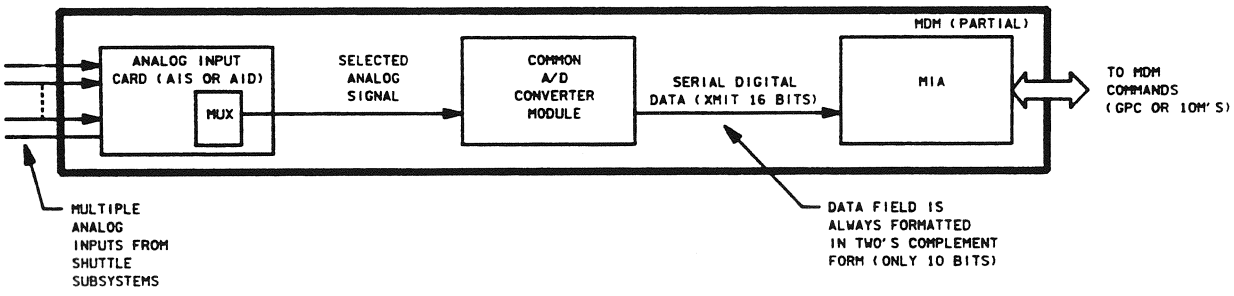
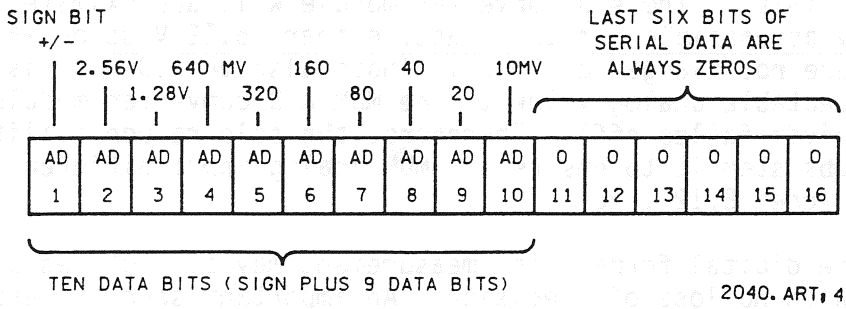


Figure 13-8.- MDM analog input signal processing.

1. Regardless of the type of analog input card connection, the MDM's common A/D converter will always convert the selected analog signal to a serial data stream of 10 bits expressed in two's complement form with a LSB equivalent to 10 millivolts dc. Thus, it follows that the maximum differential voltage which can be converted is 5.12 V dc:



The data field transmitted to the GPC (or PCMMU) is always 16 bits, but the last 6 bits are always zeros. Again, it is emphasized that the first 10 bits, which represent the analog input voltage, are always formatted in two's complement form.

2. There are several implications of the above statements:

   a. The maximum positive value that can be transmitted is equal to 0111111111.

   b. The maximum negative value that can be transmitted is equal to 1000000000.

   c. Since the LSB output of the A/D converter corresponds to 10 millivolts dc, the corresponding maximum analog voltages are:

      +MAX = 0111111111 = +5.11 V dc

      -MAX = 1000000000 = -5.12 V dc

For an AID or an AIS card, the analog input voltage may range from -5.12 V dc to +5.11 V dc, with the maximum differential voltage of 5.12 V dc. The AIS card does not provide common mode noise rejection, however.

One additional consideration will now be discussed, A/D conversion error checks:

A. The MDM design has built-in checks to detect A/D converter hardware failures (selected single-point failures do exist, however).

B. The analog input card (AID or AIS) has some feedback to the MDM processing logic as to the validity of the input data. Card hardware failures, however, are possible which permit the wrong analog signal to be selected (MUX failure) or the wrong value to be sensed (input receiver failure). The common A/D converter module, of course, has no

way of detecting such failures. The responsibility of detecting such erroneous data is the responsibility of the applications software Redundancy Management (RM).

C. Finally, the important point to make is that the MDM makes no error checks to determine if the analog input voltage exceeds the maximum differential limits. The A/D converter module will approximate the value as closely as possible (in this case, either -5.12 V dc or +5.11 V dc) and data are not flagged as invalid (note also that 0 V dc is a perfectly acceptable analog value to the MDM A/D converter module if the input receiver failed off). Therefore, the sole responsibility of the Orbiter subsystem is to ensure the MDM analog input differential voltage does not exceed 5.12 V dc.

Once in the digital format, the measurement may be shuffled around the avionics with no loss of precision. An important special case is the precision reduction in telemetry of OI measurements. For economical reasons, only 8 of the 10 bits are telemetered in the downlink bandwidth. If the measurement is unipolar, the sign bit and LSB are discarded, providing a range of 0 to 255 PCM counts. If bipolar, the two LSBs are discarded providing a range of -128 to +127 PCM counts. Flight-critical and PL MDM analogs are downlisted packed in a 16-bit halfword and, therefore, have no loss of precision.

13.6.6 Cal Curve[1]

In order to put this many-translated PCM count value to work, we must somehow convert the value into decimal EUs (the original subsystem parameter) and present it on a CRT so the crew can understand and use it. This process is conventionally called "calibration," although "scaling" would perhaps be more descriptive. The Master Measurement Data Base (MMDB) stores a "cal curve" or set of polynomial coefficients which form a function to derive EUs (EU: psia, volts, %, etc.) from a domain of telemetered PCM counts.

$$EU = A_0 + A_1(PCM) + A_2(PCM)^2 + \ldots$$

A_0, A_1, A_2 ... : polynomial coefficients

The MMDB has a maximum of six such coefficients. Since the cal curve assumes the telemetered format, unipolar OI measurements will be calibrated assuming a 0 to 250 count domain (OI bipolar assumes -128 to +127).

These cal curves are only an approximation derived by polynomial curve-fitting to data points that were derived by comparing measurement performance to a laboratory standard. A typical concern is those transducers which have a logarithmic characteristic or are desired to be displayed in a logarithmic EU (e.g., decibels). Therefore, the approximation will have some inherent error. This error is supposed to fall within the accuracy band.

13.6.7 Cal Curves in Flight Software[4,7]

Since the OI cal curves are usually defined in the 0 to 250 count domain, we must perform a computation on these coefficients to use them in SM2 (SM2 uses the 0 to 500 domain):

$$\frac{PCM_{SM}}{2} = PCM_{TM}$$

$$EU_C = A_0 + A_1(PCM_{SM}/2) + A_2(PCM_{SM}/2)^2 + \ldots$$

$$= A_0 + (A_1/2)(PCM_{SM}) + (A_2/4)(PCM_{SM}) + \ldots$$

Therefore -- $A_{0SM} = A_0$, $A_{1SM} = A_1/2$,
$A_{2SM} = A_2/4 \ldots$

$$A_NSM = A_N/2^N \text{ (unipolar)}$$

Bipolar OI measurements will work out to the following:

$A_{0SM} = A_0$
$A_{1SM} = A_1/4$
$A_{2SM} = A_2/16 \ldots$

$$A_NSM = A_N/4^N \text{ (bipolar)}$$

These equations reflect the usage of a "hybrid" Taylor expansion series. The reasons behind choosing the Taylor series for the curve fitting approximations are as follows:

A. Ease of implementation into an automated process.

B. For the precision necessities (remembering accuracy can only be kept to ±10 mv), the Taylor series is accurate enough.

C. Historically, use of the Taylor series is the standard procedure.

Both SM2 and BFS SM have the arbitrary limitation that no cal curve shall exceed third order (no more than four coefficients). Where the MMDB recommends a fourth or fifth order cal curve, Rockwell has derived (by another curve-fitting process) an acceptable third order approximation. In some of these cases, inaccuracy at the extremes has been accepted to get high accuracy in a particular region of interest. GNC application processing in both PASS and BFS includes many cal curves to allow use (for display and computation) of analog measurements.

13.6.8 Why Cal Curves Change

Hopefully, with a relatively stable vehicle configuration, the cal curve complement would not change; this is indeed a goal of the program.

The recent experience, however, has not justified this confidence. Cal curves have changed for the following reasons:

A. The cal curve in the MMDB was incorrect when SM2, BFS, or GNC was updated; consequently, the error was not discovered until it was too late.

B. Measurements change the cal curve characteristics with age or stress. The current program philosophy leaves many of these measurements on the vehicle until it is convenient to replace them (perhaps several missions later).

C. Measurements with serial number-unique cal curves are replaced with different serial numbers or types, causing an incorrect correlation.

D. In some cases, the initial documentation of a cal curve is literally a rough estimate (based on spec range) to be updated later by actual data.

As a consequence of those potentials, any cal curve should not be taken to be infallibly correct.

When a cal curve changes, the update takes a while to cycle through the system to all users; the varying update schedules can cause one user system to fall behind the others. Some of the systems of concern are:

A. Onboard

SM2 (auto update by MMDB at build time)
BFS SM (manual update "as required")
GNC (PASS and BFS) (I-loads or code "as required")

B. Ground

MCC (auto update by MMDB at build time)
SMS models (auto update by MMDB (from MCC) at build time)
- Models are in EUs which must be converted to PCM counts for FSW and telemetry.

KSC (auto update by MMDB at build time)

As a consequence, any two systems can disagree and, as long as cal curves change, will have a high probability of exhibiting transient discrepancies.

13.6.9 Offscale High and Offscale Low[6]

Many times dealing with the measurement at the extremes of its range or capability is necessary. Some of the reasons include:

A. Verifying that a Table Maintenance Block Update (TMBU) limit change is within the allowable (SM-acceptable) range of the measurement

B. Inhibiting an annunciation by adjusting the limits to extreme (SM-acceptable) values

Since the system discussed thus far has a variety of ranges, a need to specify the limits exist:

A. The "measurement range" in the MMDB is only a rough requirement and should be used only as a general description of capability. In many cases, this differs from the usable range by up to 15 percent.

B. The "functional range" of a measurement is usually considered to be -5.00 to +5.00 V output of the signal conditioner (0.00 to +5.00 V for unipolar OI measurements). Conventionally, onboard CRT display conventions generate status indicators for analog measurements when these values are reached: "L" or Offscale Low (OSL) and "H" or Offscale High (OSH). The Mission Operations Computer (MOC) in the MCC generates them at the 0 and 250 count values for OI measurements and also suppresses display of the current measurement value.

   The OI DSCs supposedly "limit" the output to the MDM to these 0 to 5.00 V values (saturating at values out of these limits). Other signal conditioners do not perform any limiting.

C. The "MDM range" of a measurement of -5.12 to +5.11 V or -512 to +511 counts is the widest range that can ever be seen by a GPC from an analog card; therefore, for TMBU purposes, this range will be used as the definition of OSL and OSH.

The SM and BFS TMBU designs incorporate range checks which will reject some FDA limit changes outside of the MDM range. Also, finite precision arithmetic in the Shuttle Data Processing Complex (SDPC) and SM can add or subtract a minute error to the desired value. Occasionally this can cause an extreme value to be rejected while the value is still valid (being "just over or under" the allowable range). Furthermore, SM incorporates two different ways of processing analog measurements. In order to work within these problems, the following rules have been developed for identifying the largest values which should be uplinked. Additionally, these guidelines help to define the useful measurement range:

SM2:

Nonlinear or negative slope linear cal curve

$[A_2 \neq 0$ or $A_3 \neq 0$, or $(A_1 < 0$ and $A_2 = A_3 = 0)]$

OSL = EU corresponding to -513 counts
OSH = EU corresponding to +512 counts

Positive slope linear cal curves

$[A_2 = A_3 = 0$ and $A_1 > 0]$

OSL = EU corresponding to -511.75 counts

OSH = EU corresponding to +510.75 counts

13.7 SPECIAL PROCESSES[6]

A number of processes are required to be accomplished by SM which cannot be accomplished by the standard processing available. These processes are primarily special control sequences involving commands and feedbacks, or simple computations to convert or combine several measurements into a more useful form. All of the special processes (fig. 13-9) are dispatched by a module called the Special Processes Executive (SSP), which spreads their executions over a period of two major cycles (1.92 seconds). This timing scheme gives most processes a single execution in this period. A number of service routines are included as special processes; e.g., Special Processes Data Out (SSO - a service routine) is the module which prepares and initiates the outputs to PL MDMs.

There are several characteristics common to all special processes:

A. Commfaulted input data are used anyway. If a value generated by the process is used elsewhere in software, that resultant value is flagged commfaulted and the user responds accordingly (displays will present an "M").

B. Any required real-time "adjustments" to the process are implemented as special processes constants which are updatable by TMBU uplink or table maintenance SPEC item entries. In some cases constants exist which are never updated. In other cases, additional constants would be very useful, resulting from a history of preliminary designs implemented without benefit of operational experience. These constants, located in the various Common Data Table (CDT) compools are initialized by the SM preprocessor.

C. Outputs to the Orbiter subsystems are routed through the PL MDMs by SSO. The sequence sets bits in the Special Processes Output Buffer (SPOB) compool which SSO cyclicly outputs.

D. While the SM design attempts to locate all annunciation within FDA, some annunciation is requested directly by the special processes. This is accomplished by setting bits in the SPOB compool. SSO will cyclicly examine these bits and request annunciation, as appropriate.

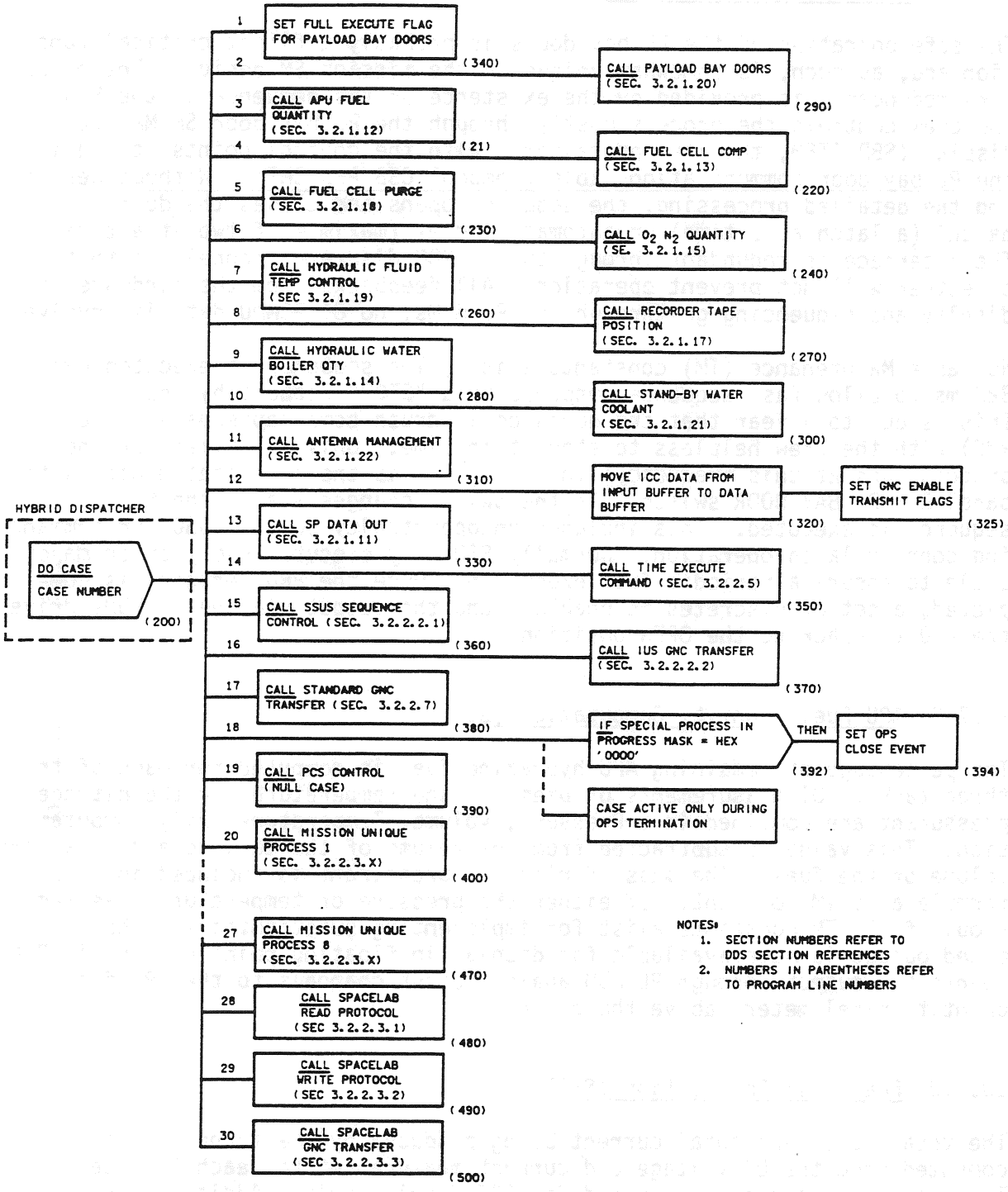


Figure 13-9.- SM Special Processes6.

13.7.1 Payload Bay Doors (SSB)

The safe operation of the PL bay doors is probably a flight critical function and, as such, is somewhat unique in the mission SM design. The necessary redundancy is provided by the existence of the sequence in the BFS. The crew controls the process mostly through the PL bay door SM MM 202 display (SBD_ITEM, the item processor), with the control points stored in the PL bay door communication table (Compool CSB_PBD_CMT). Without describing the detailed processing, the sequence opens and closes the doors in manual (a latch at a time) or automatic modes (maximum of two at a time). The interface is redundant through the PL MDM discrete channels so that loss of either will not prevent operation. All feedbacks from the hardware for display and sequencing go through the PL MDMs; no OI PCMMU data is involved.

No Table Maintenance (TM) constants exist. The sequence is executed every 320 ms to allow fast sequence response to a "STOP" request by the crew. This is due to a fear that the doors could crush some "appendage" (i.e., RMS) with the crew helpless to stop it in time. Only a fragment of the process runs at this frequency; that fragment is the part that monitors the panel R13 PL BAY DOOR switch. If the switch changes state, the full sequence is executed. This includes an opportunity to call SSO for commanding door or latch operation. Normally SSO only executes every other major cycle to ensure a periodic full execution. Once the PBD sequence is completed, a set of discretes is enabled, and the output via the PL MDMs drive the PBD talkback to the OPEN position.

13.7.2 APU Fuel Quantity Computation (SSA)

The percentage of remaining APU hydrazine fuel is computed for each of the three tanks. OI measurements of pressure and temperature for the nitrogen pressurant are combined in a Pressure, Volume, Temperature (PVT) computation. This value is subtracted from the volume of the tank to arrive at the volume of the fuel. The mass of nitrogen pressurant is included in this formula as a TM constant. If either the pressure or temperature measurement should fail, TM constants exist for implementing substitutions. The computed output is made available for display in floating point EUs and in PCM counts for output through PL MDM analog output channels to the APU fuel quantity panel meters above the eyebrow.

13.7.3 Fuel Cell Computation (SSF)

The total power and total current being produced by the three fuel cells is computed from the OI voltage and current measurement for each fuel cell. These two products are generated for CRT display only. Additionally, three bilevels are generated for each fuel cell which coarsely indicate how much power that fuel cell is producing; these are used as FDA preconditions on each fuel cell stack temperature which normally varies with power level. Two TM constants define the three power ranges for the stack temp preconditioning.

13.7.4 Fuel Cell Purge Sequence (SSC)

This sequence accomplishes a fuel cell purge when requested by the crew. The proper configuration of one PL MDM discrete input and several OI discretes will cause the sequence to command the purge line heaters on via PL/MDM outputs. The sequence also monitors three OI line temperatures to verify heater performance. If the lines warm up, each fuel cell properly configured will be purged for Y minutes (typically 2) by PL MDM purge commands. This "Y" value can be changed via SPEC 60. This sequence incorporates a closed loop check which compared reactant flow to computed flow (based on fuel cell produced current) to verify purge valve operation. Whenever reactant flow is greater than the computed flow, the timer is inhibited and the purge executes by opening the fuel cell purge valve. After a selected interval, the purge line heaters are turned off, completing the purge.

Only one TM constant exists; this constant is the Y-minute purge timer discussed above and common to all three fuel cells. When the timer expires, the valve is commanded closed, and a valve-closed verification is performed.

13.7.5 O_2, N_2 Quantity Computation (SSN)

OI pressure and temperatures are combined to compute the mass (in lbs) of each of the emergency O_2 and the two gaseous N_2 tanks.The output is only used for CRT display. The computation (using a PVT relationship) is executed at least once per 2 seconds. No TM constants exist for this process.

13.7.6 Hydraulic Fluid Temp Control Circulation Pump Sequence (SST)

The early days of the program held concerns about the survivability of the hydraulic system in the cold vacuum of space. To safeguard the system, this sequence utilizes the circulation pumps to move cold hydraulic fluid from cold areas to warm areas. Since each operating circulation pump uses 2000 watts, minimizing the usage time of these three pumps is essential.

The sequence monitors $\approx$50 OI temperature measurements distributed over the lines of the three hydraulic systems. If one temperature is below a low limit (unique TM constant), the appropriate circulation pump is run until either the temperature gets above a high limit (TM constant) or another hydraulic system has a cold temperature and the current system's timer (TM constant) has run out. Auto control using the sequencing software shall be enabled only for the hydraulic systems whose respective circulation (circ) pump control panel switch is in the GPC position. Reading of the switch position is redundantly performed to ensure accuracy. Only one circ pump shall be on at a time for an out-of-limits low reading. Pumps are commanded on and off via PL MDM commands. OI switchscans indicate which of the three hydraulic systems is eligible to be circulated. The sequence monitors the OI circulation pump outlet pressure to annunciate if the previous pump did not turn off before the next pump is commanded on. Preconditioned FDA monitoring is used to detect and annunciate a failed-off pump.

The operation thus described is called "thermostat mode". If the range (low and high limit) of the TM constants is widened in each eligible hydraulic system (each system is always "cold"), the sequence operates in a default "timer mode", cycling each eligible system for a defined period (the timer TM constant). To monitor the overall effectiveness of the sequence, the ≈50 OI temperature measurements are individually monitored by FDA to annunciate if dangerous temperatures are impending. The FDA limits are independent from the sequence TM constant limits; this may be a source of confusion. To update FDA, use the MSID of the limit sensed OI measurement (a V58T-MSID) as usual. To update a sequence limit, use the MSID of that limit; a pair of V92T-MSIDs correspond to each V58T measurement.

The circulation pumps are occasionally operated for reasons other than thermal concerns. In these cases, the sequence operation may be effectively altered by updating the TM constants.

13.7.7 Recorder Tape Position (SSR)

The three recorder tape position measurements, which go through the PL MDM's, are packed in an unorthodox way. The LSB appears as the leftmost bit in the 5-bit word. Since that standard scaling (from PCM counts to EUs) assumes the reverse, this value must be specially converted to a useful percentage readout (i.e., EUs), and that is all that this process does. The conversion is by table reference, and this computation is performed at least once every 2 seconds.

13.7.8 Hydraulic Water Boiler Quantity (SSH)

This process computes the remaining water in the three hydraulic water boiler water tanks. This water is used to cool the hydraulic fluid and APU lube oil. This process is exactly like APU Fuel Computation (SSA - section 13.7.2) except that: no pressure or temperature substitutions (TM constants) exist, and the PL MDMs feed the cockpit meters and CRTs.

13.7.9 Standby Water Coolant Loop Temperature Control Sequence (SSS)

On orbit, only one of the two water coolant loops is actively used. There is some concern that the inactive loop will freeze in cold thermal conditions. For this reason, this sequence will periodically command all three water loop pumps on via PL MDM commands. Only the pump selected for GPC operation will actually respond to the commands. The sequence is a simple timer, with the pump on and off times being TM constants. The sequence does not use any feedbacks, although the FDA monitoring of the water loops does use the sequence commands as part of the preconditioning.

13.7.10 Antenna Management (SSM)

One of the tasks SM has is selecting the correct S-band antenna or pointing the Ku-band dish antenna correctly to ensure communication with the ground. The process relies on Orbiter state vector information (ICC'ed from the GNC GPCs) to analyze the position and attitude of the vehicle relative to the Earth and the Tracking and Data Relay Satellites (TDRSs). The commanding to the S-band antennas is through PL MDM discrete outputs, with each MDM dedicated to one of the two S-band "strings." Feedbacks are received through the PL MDMs, with a few moding discretes coming from the OI PCMMU. The Ku-band dish antenna is controlled and monitored through a dedicated serial I/O channel (card 8, channel 2) of MDM PF1. The crew cannot manually control the dish well enough to provide usable communications. Numerous TM constants exist.

13.7.11 Special Processes Data Out (SSO)

See section 3.3.3.4 PL MDMs (paragraph E).

13.7.12 Time Execute Command (TEC) Buffer Poll (PMQ)

The SM uplink process (SUL) can receive uplinked time execute commands which are to be output to the subsystems at a specific uplinked GMT. The outputs may go to standard serial output channels (PL, flex MDM, or dual port outputs), Spacelab serial channel (through the PL MDMs), or outputs to the PSP. Discrete outputs of this sort are handled by the Orbiter standard uplink process of Stored Program Commands (SPCs). The output processing is overseen by PMO (standard output control).

13.8 MISSION SELECTABLE PROCESSES[6,7]

Mission Selectable Processes (MSPs) are unique, but can be used on numerous missions; these MSPs are controlled by the Payload processing control indicators (PCIs). These indicators can be turned OFF and ON by cargo control (CC) SPEC item entries, by other MSPs, by payload control supervisor (PCS) sequences, or OPS initialization. Examples of MSPs are listed in Table 13-VII below, with descriptions of several MSPs following.

TABLE 13-VII.- MISSION SELECTABLE PROCESS CONTROL DATA

Process name	Process ID	Requirements section	Rate	Process[1] ACT/DEACT indicator	ACT/DEACT method				
					OPS	PCS	CC	MUP	MDP
SPACELAB SERIAL DATA INTERFACE	000001	7.1.1	3.125 Hz		X				
SSUS SEQUENCE CONTROL PROCESS	000002	7.1.2	1.04 Hz	V92X1217X			x		
IUS GN&C DATA TRANSFER	000003	7.1.3	.52 Hz		X				
SPACELAB SPECIAL PROCESSING	000004	7.2.1	.52 Hz		X				
MASS MEMORY PATCH	000005	4.8	DEMAND		X				
SUBSYSTEM CONFIGURATION MGMT	000006	4.3	DEMAND		X				
STANDARD GN&C DATA TRANSFER	000007	4.11	.52 Hz		X				
PAYLOAD CONTROL SUPERVISOR	000008	4.12	1.04 Hz		X				

[1]If no MML is specified, there is no PAD for the associated process.

13.8.1 SSUS (PAM) Sequence Control (PDS)

While most PAM deploy monitoring is done by FDA, and control is accomplished by crew item entries, the actual final sequence start and deploy commands are issued under control of this sequence. The sequence uses no inputs from the hardware but does access the deploy time. The sequence is activated by item entry on a CC SPEC (usually the Deploy SPEC). The sequence also generates a number of reference time discretes which represent an ongoing estimate of where the hardware "mechanical" sequence should be. This estimate is based on the countdown clock. These discretes are used as preconditions for FDA monitoring of the mechanical sequence. Countdown clock reference times, final sequence start, SSUS ID, and the SSUS Deploy time are all TM constants. The two commands are issued by a call to standard output control (PMO).

13.8.2 IUS GNC Transfer (PDL)

This process gathers Orbiter state vector and attitude data received from GNC by ICC, reformats the data as appropriate, and outputs the data via PSP or to the IUS Communications Interface Unit (CIU) via an MDM SIO channel. PSP outputs are implemented by call to the PSP reserve buffer (SPR). Outputs through a CIU are implemented by call to Standard Output Control (PMO). The outputs occur under control of PCIs. The data is processed as a time consistent data set so that parameters from two different ICC updates are not used in the same IUS transfer.

13.8.3 Standard GNC Data Transfer (PMG)

This process is basically the same as the IUS GNC transfer except:

A. The data are not specially formatted for the IUS.

B. Up to eight transfer combinations may be specified by the preprocessor, each of which is controllable by PCI.

13.8.4 Spacelab Serial Data Interface (SLSIO)

The SM GPC has the ability to exchange serial data with each of the two Spacelab computers, the subsystem computer (SSC) and the experiment computer (EXC). Each Spacelab computer is assigned two independent MDM serial channels as follows:

Computer	Chan	MDM
SSC	A	PF1
SSC	B	PF2
EXC	A	PF1
EXC	B	PF2

Only one channel is active at a time.

TABLE 13-X.- ALPHABETICAL LISTING OF THREE CHARACTER PROCESSES[6]

Name	Process
DMP	MASS MEMORY MESSAGE PROCESSOR
PC1	CARGO CONTROL SPEC FUNCTION CONTROL SEGMENT 1
PC2	CARGO CONTROL SPEC FUNCTION CONTROL SEGMENT 2
PCI	CARGO CONTROL INIT/CLEANUP
PCK	CARGO CONTROL ITEM PROCESSOR
PCL	CARGO CONTROL STORED SERIAL COMMAND
PCM	CARGO CONTROL DISCRETE ITEM PROCESSOR
PCP	CARGO CONTROL PSP COMMANDS
PCR	CARGO CONTROL HARDWARE PULSED COMMANDS
PCU	CC SPACELAB CMD REQUEST
PCW	CARGO CONTROL SOFTWARE PULSED COMMANDS
PDL	IUS/GN&C TRANSFER
PDS	SSUS SEQUENCE CONTROL
PMC	CYCLIC STANDARD OUTPUT
PMG	STANDARD GN&C TRANSFER
PMO	STANDARD OUTPUT CONTROL
PMP	SPACELAB RESERVE BUFFER PROCESSOR
PMQ	TEC BUFFER POLL
PMR	SPACELAB READ PROTOCOL
PMT	SPACELAB GN&C DATA TRANSFER
PMW	SPACELAB WRITE PROTOCOL
PPC	PCS SPEC FUNCTION CONTROL SEGMENT
PPF	PCS SEQUENCE CONTROLLER
PPI	PCS I/O PROCESSOR
PPK	PCS ITEM PROCESSOR
PPM	PCS MASS MEMORY READ PROCESSOR
PPN	PCS CONTROL PROCESSOR
PUS	SPACELAB SPECIAL PROCESSING
RAS	AUTOMATIC SEQUENCE PROCESSOR
RBM	HARDWARE FAULT ANNUNCIATION FILER
RCD	MCIU DECODER
RDD	DEDICATED DISPLAY
REX	RMS EXECUTIVE
RFP	POSITION HOLD

TABLE 13-X.- Continued

Name	Process
RHM	HEALTH MONITOR
RIT	TEMPERATURE PROCESSOR
RJS	HAND CONTROLLER
RKG	KINEMATIC DATA GENERATOR
RMC	RMS SPECIALIST FUNCTION CONTROL SEGMENT
RNC	MCIU ENCODER
RPO	RESOLVED POSITION ALGORITHM
RQC	INPUT PROCESSING AND CONFIGURATION DETERMINATION
RRP	RESOLVED RATE PROCESSOR
RSC	SINGLE JOINT CONTROL
RTV	TOTAL VELOCITY
RUD	RMS ITEM PROCESSOR
RVM	CONSISTENCY AND ENCODER CHECKS
RWP	POSITION HOLD CHECKS
RXY	CONFIGURATION INITIALIZATION
RYE	DATA CONVERSION PROCESSOR
SAC	PCMMU CLEAR CHAIN
SAF	SM DATA ACQUISITION/FDA
SAM	ANTENNA MANAGEMENT ITEM PROCESSOR
SBC	SM BASIC CLEANUP
SBD	PAYLOAD BAY DOORS ITEM PROCESSOR
SBI	SM BASIC INITIALIZATION
SBS	BACKWARD SCALING
SCI	SCM INITIALIZATION/CLEANUP
SCK	SM CHECKPOINT PROCESS
SCM	SUBSYSTEM CONFIGURATION MANAGEMENT PROCESS
SCO	SCM GROUND C/O SPEC CONTROL SEGMENT
SCS	SM SPECIALIST FUNCTION CONTROL SEGMENT
SDT	SM HYBRID DISPATCHER TABLE UPDATE
SFL	FORMAT LOAD PROCESS
SGC	GROUND C/O ITEM PROCESSOR
SLS	PCMMU/PL COMM SPECIALIST FUNCTION CONTROL SEGMENT
SM2	ORBIT/DOORS (OPS2) CONTROL SEGMENT
SM4	SM PAYLOAD (OPS4) CONTROL SEGMENT

TABLE 13-X.- Concluded.

Name	Process
SPC	PCMMU/PL COMM PROCESS
SPE	PSP REJECTION PROCESSING
SPN	INTERPROCESS PRECONDITION CONTROL
SPP	PRECONDITION PROCESSING
SPR	PSP RESERVE BUFFER
SPS	PSP CYCLIC
SRE	SM RESTORE PROCESS
SSA	APU FUEL QUANTITY
SSB	PAYLOAD BAY DOORS
SSC	FUEL CELL PURGE
SSF	FUEL CELL COMPUTATION/FUEL CELL END CELL HEATER MONITOR
SSH	HYDRAULIC WATER BOILER QUANTITY
SSM	ANTENNA MANAGEMENT
SSN	O_2/N_2 QUANTITY
SSO	SPECIAL PROCESSES DATA OUT
SSP	SPECIAL PROCESSES EXECUTIVE
SSR	RECORDER TAPE POSITION
SSS	STANDBY WATER COOLANT LOOP TEMPERATURE CONTROL
SST	HYDRAULIC FLUID ON-ORBIT TEMPERATURE CONTROL
STC	TABLE MAINTENANCE CYCLIC PARAMETER UPDATE
STM	TABLE MAINTENANCE PROCESS
STS	TABLE MAINTENANCE SPECIALIST FUNCTION CONTROL SEGMENT
SUL	SM UPLINK PROCESSOR
S2I	OPS 2 INITIALIZATION/CLEANUP
S4I	OPS 4 INITIALIZATION/CLEANUP
VMM	MASS MEMORY READ/WRITE SPEC CONTROL
VMP	MASS MEMORY READ/WRITE SPEC ITEM PROCESSOR
VCY	GTS CYCLIC UPDATE PROCESSOR
VTF	TELEMETRY FORMAT LOAD PROCESS

13.9 REFERENCES

1. Computer Program Development Specification (CPDS), vol. V, book 8, Systems Management Level B Software Requirements, version: Rev G, 10-1-86, SS-P-0002-580G.

2. Data Processing System Console Handbook, version: Basic, PCN-2; 9-15-85, JSC-18819.

3. Data Processing System Dictionary, version: Generic, 4-15-88, OI-8B, JSC-17322.

4. Data Processing System System Briefs, version: Basic, Rev A, PCN-1; 9-15-85, JSC-18820.

5. Master Unit, Pulse Code Modulation, Specification, Version: 4-30-82, RI document MC476-0130.

6. Operational Detailed Design Specification (DDS), vol. III, part 2 - Systems Management, version: OI-8B, NAS9-16920

7. Systems Management Level C Flight Software Requirements, STS-26A, version: Final, 8-87, JSC-19590-26A.

SECTION 14
BFS BACKUP OPERATING SYSTEM

CONTENTS

Section		Page
14.1	OVERVIEW	14-1
14.2	ARCHITECTURE	14-1
14.3	EXECUTIVE OPERATIONS	14-2
14.3.1	Minor Cycle	14-2
14.3.2	Tracking PASS	14-4
14.3.2.1	PASS Listen Commands	14-5
14 3.2.2	PASS/BFS Synchronization	14-8
14.3.2.3	Max Rate Transfer Data Processing	14-10
14.3.3	I/O RESET Processing	14-10
14.3.4	Engage/Disengage Transition	14-14
14.3.5	Startup (HALT-STANDBY)	14-14
14.3.6	Protected Memory Checksumming	14-15
14.3.7	Application Transfer Data Inputs	14-15
14.3.7.1	1.04 Hz Transfer Data (OPS 1 and 3)	14-15
14.3.7.2	G9 One Shot Transfer Data	14-16
14.4	ERROR PROCESSING	14-18
14.4.1	I/O Error Processing	14-18
14.4.2	GPC Error Processing	14-19
14.4.2.1	Restart	14-21
14.4.2.2	Error Trap	14-28

SECTION 14
BFS BACKUP OPERATING SYSTEM

14.1 OVERVIEW

The original fifth GPC design concept included only non-flight critical functions; i.e., systems management and special processes. Backup flight critical functions were added to protect against a possible PASS generic programming error in the ascent and entry flight phases. The flight-critical software was designed to be a single string system for takeover in loss-of-control situations. Program management subsequently expanded the scope to include Return To Landing Site (RTLS), PL door open/close sequences, fast sep in first stage, and Transatlantic Abort (TAL). Restart capabilities were included to recover from power transients and transient GPC hardware problems. The BFS emerged as an integral part of the Shuttle system, satisfying the original fifth machine SM concept plus providing a PASS backup.

With the expanded BFS requirements and, in particular, the addition of BFS software requiring sensors data in parallel with PASS, the BFS Backup Operating System (BOS) became more complicated and slaved to the PASS. This systems brief will provide background information and will clarify the operating system, the tracking of PASS, and the various error detection and recovery processes.

14.2 ARCHITECTURE

The BFS software can be partitioned into Guidance, Navigation, and Control (GN&C); SM/Special Processes (SP); and Backup System Services (BSS). Further, BSS can be subpartitioned into BOS, MCDS, FDA, Dedicated Display Devices (DDD), uplink processing, PCM downlist, MTU, launch data bus processing, TFL, payload bay door processing, and limited payload support. BOS is the system executive software providing control over data inputs, job processing, and outputs. BOS also monitors for engage/disengage requests and performs the appropriate transition.

BOS operates in 40 ms increments (one minor cycle) with 24 minor cycles making up a major cycle (960 ms). The fastest input-job processing-output sequence repeats every minor cycle and is called Max Rate; i.e., 24/major cycle or 25/second (Hz). Alternate (Alt) Rate inputs-job processing-outputs repeat in multiple minor cycles; e.g., 12/major cycle (12.5 Hz), 6/major cycle (6.25 Hz), and 3/major cycle (3.125 Hz).

Application jobs in the BFS are partitioned into groups according to their frequency requirements and whether they can be suspended to perform time critical jobs. The BFS processing scheme is basically cyclic; however, less important jobs have been identified and can be suspended to perform time critical cyclic tasks. The time critical cyclic tasks are divided into Max Rate and Alt Rate. The Max Rate inputs-jobs-outputs repeat every minor

cycle and occur at specific times within the minor cycle. Alt Rate inputs-jobs-outputs repeat in multiple minor cycles. The Alt Rate inputs and outputs occur at specific times within the appropriate minor cycle, and the Alt Rate job processing follows the completion of the Max Rate job processing. The non-time critical jobs are called background jobs and are executed following Max and Alt Rate job completion. These background jobs require no data bus I/O and, therefore, can run asynchronously with respect to the Max/Alt Rate jobs.

14.3 EXECUTIVE OPERATIONS

BOS performs the executive tasks required to receive input data, process jobs, and command outputs. When errors occur in performing these tasks, BOS performs error processing to interpret the error and recover from the erroring condition. BOS is designed to recover from any erroring condition except those conditions which are permanent and cause the software flow to enter into continuous repeated recovery attempts.

In general, BOS performs two functions at the same time; i.e., I/O data bus transactions and job processing. Prior to each data bus input transaction, whether PASS or BFS commanded, BOS initializes the MSC timer to count across the input transaction. BCE programs are also started to receive each element of input data. While the input data are being received by the BCE programs, BOS can be processing jobs. BOS checks the input data status to determine if all expected inputs were received without error, and if the inputs were completed within the time allotment. If errors occur in either inputting of data or in processing jobs, BOS immediately performs error processing.

The following sections provide more insight into the BOS operating scheme, minor cycle operation, and error processing. Emphasis is placed on those BOS operations which are noticeable to the crew and ground controllers.

14.3.1 Minor Cycle

The minor cycle is 40 ms in duration and is bounded by Program Counter (PC) 1 interrupts. Each occurrence of a PC 1 interrupt marks the end of the previous minor cycle and the beginning of the upcoming minor cycle as shown in figure 14-1. The PC 1 interrupt is issued when the PC 1 clock decrements down to zero. BOS loads the clock at the beginning of each minor cycle with 39.994 ms (40 ms when the instruction execution time is included). When the BFS is in the PASS tracking mode, the PC 1 clock is adjusted in mid-minor cycle to stay in sync with the PASS redundant set. Note that when the BFS is in the engaged mode, BOS can extend the minor cycle to 50 ms (add an extra 10 ms to the PC 1 clock) when Alt Rate processing loads require more than 40 ms to complete.

Time critical events within the minor cycle are cued to the occurrence of the PC 2 interrupt which occurs when the PC 2 clock decrements to zero. The PC 1 Interrupt Handler loads the PC 2 clock three times during each BFS

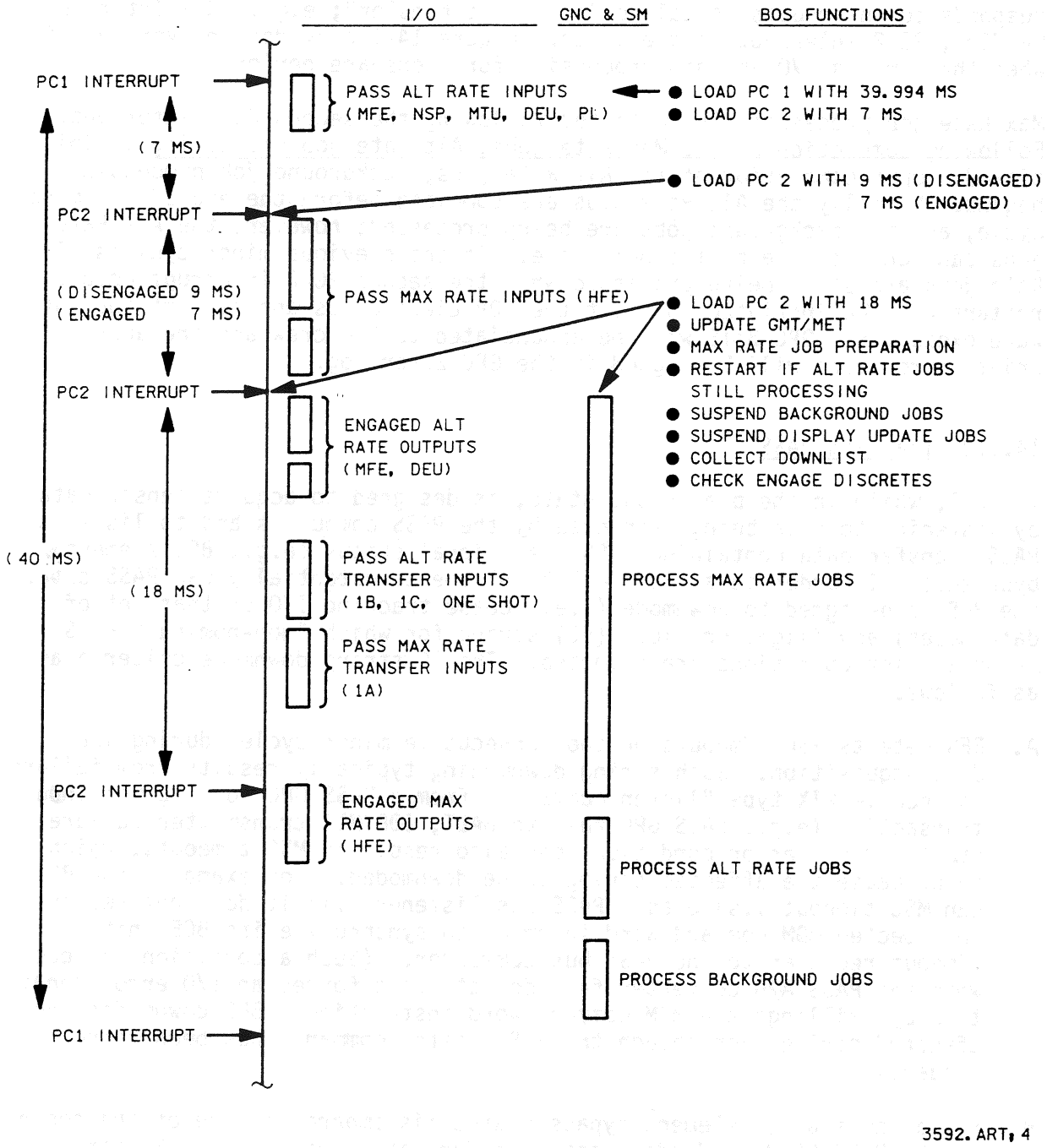


Figure 14-1.- Minor cycle overview.

minor cycle causing three PC 2 interrupts. Note that the software that responds to interrupts is called "interrupt handler"; e.g., PC 1 Interrupt Handler, PC 2 Interrupt Handler, etc. Figure 14-1 provides an overview of when the various I/O and job processing functions are performed.

Max Rate job processing is initiated following the second PC 2 interrupt. Following completion of the Max Rate jobs, Alt Rate job processing is initiated. Upon completion of the Alt Rate jobs, background job processing begins. Normally the Alt Rate jobs are complete before the end of the minor cycle, and the background jobs are being processed; however, the Alt Rate jobs can run into the next minor cycle. If the previous minor cycle's Alt Rate jobs are still being processed when the second PC 2 interrupt occurs, a restart will be initiated to reset the IOP electronics and restart the software execution. GPC BITE will be annunciated to the crew and the appropriate error code will be logged in the GPC Error Log.

14.3.2 Tracking PASS

The BFS, while in the pre-engage state, is designed to acquire sensor data by listening to data being retrieved by the PASS computers and to listen to PASS transfer data containing PASS operational status (e.g., BCE element bypasses). In order to avoid "pollution" due to potentially bad PASS data, the BFS is designed to downmode (i.e., cease tracking I/O on that set of data buses) any flight critical (FC) string for which "non-nominal" PASS redundant set operations are detected. The FC string downmode criteria are as follows:

A. BFS detects MSC timeouts on two consecutive minor cycles during input data acquisition. Such string downmoding typically results from failure to receive WIX type "listen commands" from a PASS GPC for a given input transaction (e.g., PASS GPC fail to HALT, IOP MIA transmitter failure, etc.). Other error conditions can also result in MSC timeouts, which in turn, cause the affected string to be downmoded. For example, the BFS can MSC timeout...same as a PASS bus listener...if it does not receive an expected MDM command word in order to synchronize its BCE initial timeout register to the PASS bus commander. (Such a condition can occur when the PASS A/D converter fail detect logic forces an I/O error condition by "killing" the MDM command word instruction. BFS downmodes the affected string even though the BFS "listen command" has been correctly issued.)

B. BFS detects a BCE element bypass status miscompare for one of the copies of the PASS 25 Hz redundant set transfer data (one copy per FC string, transferred on data buses FC5 through FC8).

C. The BCE element bypass status contained within the PASS XFER data differs with respect to the corresponding BFS element bypass indicators (i.e., either BFS or PASS has bypassed one or more BCE elements not bypassed by the other software system). It should be noted that the BFS does not maintain an active bypass status for either star tracker (STKR) or microwave landing system (MLS) elements.

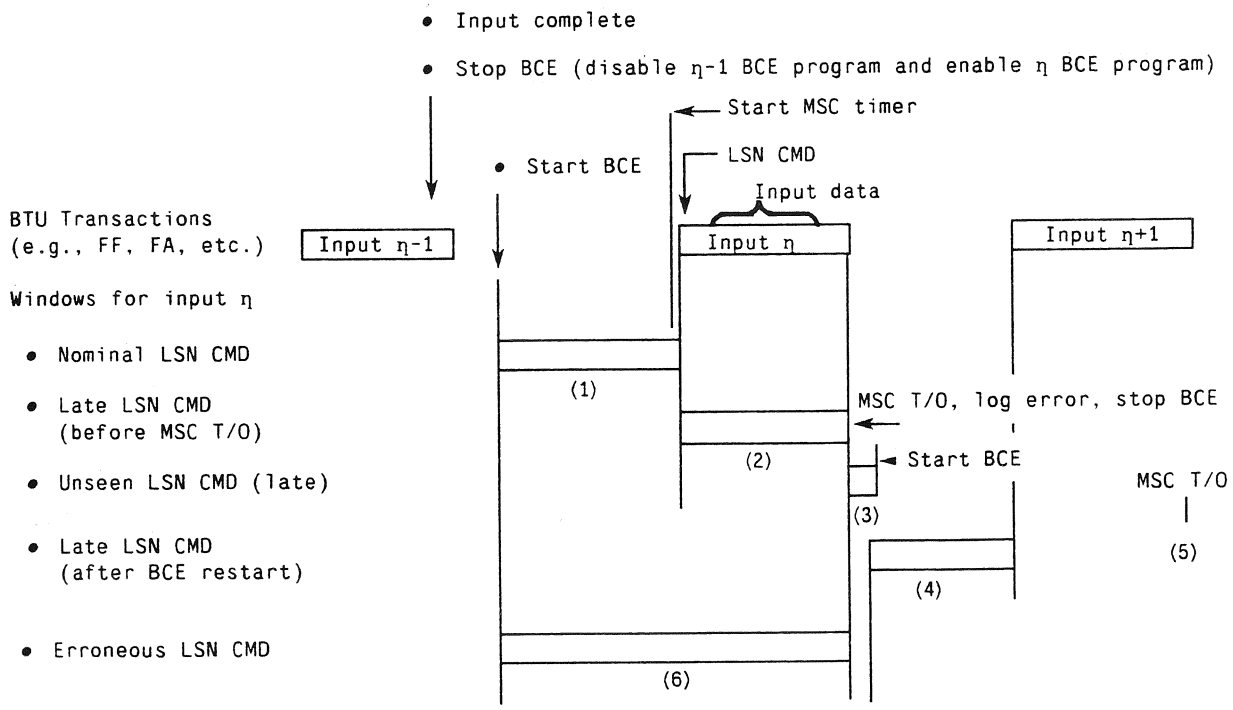
D. BFS tracking less than two strings; e.g., BFS is tracking strings 1 and 2 and string 1 LSN CMD's cease - BFS downmodes both strings. Resultant BFS state is referred to as "standalone".

14.3.2.1 PASS Listen Commands

All input transactions of flight critical MDM, EIU, NSP, and MTU data initiated by PASS are preceded by WIX type LSN CMD's containing information (Interface unit address (IUA) and a BCE program identification index) that the BFS requires to start the correct BCE program for receiving the incoming data. Table 14-I presents the LSN CMD's contained in the PASS input I/O profiles. The BFS, in order to avoid pollution, performs several LSN CMD checks to make sure the PASS computer commanding the bus is operating as expected. These checks involve testing to see if the LSN CMD is the one expected and if the LSN CMD arrived during the expected window. Figure 14-2 highlights the various LSN CMD windows and the following paragraphs discuss the BFS response.

TABLE 14-I.- BFS LISTEN COMMANDS

PASS I/O profile	LSN CMD index	Data bus BTU	LSN CMD IUA
HFE input (24/MC)	0	EIU 1	17
	2	2	23
	4	3	24
	6	FF 1-4	10
	6	FA 1-4	12
NSP read (6A/MC)	8	NSP 1(2)	10
MFE input (6B/MC)	10	FF 1-4	10
	10	FA 1-4	12
MTU read (1B/MC)	12	MTU 1-3	10
HFE output XFER data (25.0 Hz)	14	BFS	5
(1.04 Hz)	16	BFS	5
(1.04 Hz)	18	BFS	5



NOTES:
(1) LSN CMD received early enough for data to complete before MSC timer expires.
(2) LSN CMD received; however, data acquisition does not complete before MSC T/O; i.e.,MSC T/O logged against input η.
(3) LSN CMD arrives after MSC T/O (while BCE is momentarily stopped) and is not seen; i.e., MSC T/O logged against input η.
(4) LSN CMD arrives after MSC T/O and BCE reinitialized looking for next LSN CMD (η+1). BCE branches to illegal instruction and stops, causing next LSN CMD to be missed. MSC T/O logged against input η.
(5) LSN CMD/inputs (η+1) are not seen because BCE is stopped. MSC T/O is logged, with illegal instruction in status word, against input η+1.
(6) BCE stops on illegal instruction. MSC T/O is logged with illegal instruction against input η.

18820*024

Figure 14-2.- BFS listen command windows (typical).

Flight critical input transactions (i.e., HFE, MFE, NSP, and MTU) are composed of one to four BTU transactions per data bus; e.g., the HFE input transaction on the FC 8 bus contains three EIU BTU transactions and one FA4 BTU transaction. Each BTU transaction is composed of one or more input elements; e.g., FA4 return word, R OMS Pc, and PROM Seq 3-10 (D). Also, three XFER data transactions from PASS to BFS occur only on the aft FC buses; i.e., 25 Hz PASS element bypass status (every minor cycle), 1 Hz PASS state vector data (minor cycle 15), and 1 Hz PASS calibration data (minor cycle 23). Each of these flight critical BTU and XFER transactions is preceded with a LSN CMD. PASS also transmits a LSN CMD prior to each DEU poll response which cues the BFS to start the DK bus poll response BCE program.

NOMINAL LSN CMD - The BFS does not specifically verify that the LSN CMD is received at the correct time but indirectly makes the verification by checking to see if the BTU transaction completes before the MSC timer expires. As shown in figure 14-2, the nominal LSN CMD can arrive as early as immediately following the startup of the BCE after the previous transaction, and also, the entire BTU transaction can be complete before the BFS MSC timer is started. The LSN CMD normally arrives during the latter part of the nominal LSN CMD window and the transaction normally completes immediately prior to the MSC timer expiring. Note that for the HFE input transaction, the BFS specifically determines when the first LSN CMD (EIU 1) arrives on the aft FC buses and adjusts the minor cycle time for staying in sync with the earliest PASS computer.

LATE LSN CMD (BEFORE MSC T/O) - When the PASS is tardy in performing an input transaction, the transaction will complete successfully in PASS (MSC timer is also started late); however, the BFS MSC timer is started at the normal expected time (not knowing that the PASS is tardy) and will time out before the input transaction completes, and therefore, count one error against the input transaction. If two consecutive errors occur (e.g., FF2 LSN CMD's) on an input transaction (e.g., HFE), the BFS will downmode the entire string.

UNSEEN LATE LSN CMD - Following each BTU transaction, the BCE program is stopped, error logging performed as appropriate, status registers cleared, the previous BCE program disabled, the next expected BCE program enabled, and the BCE restarted looking for the next expected LSN CMD. If PASS should happen to transmit a late LSN CMD during the BCE stop period, the BFS will never see the command and consequently will not log any additional error condition other than the MSC T/O caused by the transaction not completing on time.

LATE LSN CMD (AFTER BCE RESTART) - When the PASS I/O is so late (due to excessive processing) that the expected LSN CMD arrives after the BCE has been restarted looking for the next expected LSN CMD, then two I/O errors are logged; i.e., one MSC T/O because the last input transaction did not complete on time, and one MSC T/O on the next input transaction because the BCE is at stop waiting for error processing. Note that all BCE programs on a bus are nominally disabled except the BCE program for the next expected inputs. Consequently, a late LSN CMD for a disabled BCE program will cause

the BCE program to branch on the disable to an illegal instruction, load an illegal instruction indicator in the BCE status registers, and stop the BCE from listening for additional LSN CMD's. When the next expected LSN CMD arrives, the BFS will have started the MSC timer; however, it will not see the LSN CMD (BCE stopped) and will perform MSC T/O error processing. However, this MSC T/O will have an illegal instruction indicated in the status words instead of all zeros.

ERRONEOUS LSN CMD - Lastly, if an erroneous LSN CMD (not previous late LSN CMD) is received, the BCE again is stopped, causing the BFS MSC timer to expire without inputs complete. At this time, a MSC T/O is logged with an illegal instruction indicated in the status words; i.e., similar to a late LSN CMD.

14.3.2.2 PASS/BFS Synchronization

The BFS attempts synchronization and tracking of the PASS redundant set for the following conditions: (1) BFS GPC is moded from HALT to STANDBY/RUN (defaults to OPS 0); (2) transitioned to OPS 0, G1 or G3; (3) BFS keyboard initiated I/O RESET; and (4) recovering from a software restart. Note that for I/O RESET's, the BFS forces I/O resynchronization with the PASS only if one or more strings are downmoded.

As shown in table 14-II, the PASS transmits EIU LSN CMD's (Index 0) on data buses FC5 through FC8 at the start of the 25 Hz HFE input transaction. During tracking initialization, the BFS goes into a "listen" mode for one or more EIU 1 LSN CMD's. When one is detected (the earliest LSN CMD if more than one is detected), the PC 1 clock is updated to expire 7 ms before the next expected minor cycle EIU 1 LSN CMD's. Also, the FC 5, 6, 7, and 8 BCE programs are enabled to listen for the HFE output 25 Hz 1A transfer data LSN CMD's and to receive the transfer data. Note that all four copies of the transfer data should be identical provided all four PASS GPC's are part of the current redundant set. If two or more strings of transfer data are received and BCE element bypass status compare bit-by-bit, and if EIU 1 LSN CMD's are received on the next minor cycle, PASS synchronization is considered established on those strings. The final steps in syncing and tracking PASS are to match the PASS major/minor cycle and then upmode the input elements on those strings the BFS is tracking to match those identified in the transfer data. Note that all input elements corresponding to strings not being tracked by BFS will be bypassed.

Once synchronization with PASS has been established, the process of tracking LSN CMD's and comparing 25 Hz 1A transfer data is repeated every minor cycle. When either MSC timeouts occur (typically, a LSN CMD is absent) for two consecutive minor cycles or a copy of transfer data miscompares, the appropriate string will be downmoded. If less than two strings remain, the BFS downmodes the remaining string and the BFS modes to "standalone".

TABLE 14-II.- FLIGHT CRITICAL DATA BUS INPUT PROFILE:
PREENGAGE 25 HZ (24/MAJOR CYCLE) HFE INPUTS

FC1	FC2	FC3	FC4	FC5	FC6	FC7	FC7
Delay (≤2.5 ms)	Delay (≤2.5 ms)	Delay (≤2.5 ms)	Delay (≤2.5 ms)	LSN CMD(0) EIU 1 6 words	LSN CMD(0) Delay	LSN CMD(0) Delay	LSN CMD(0) EIU 1 6 words
				Delay	LSN CMD(2) EIU 2 6 words	Delay	LSN CMD(2) EUI 3 6 words
				Delay	Delay	LSN CMD(4) EIU 3 6 words	LSN CMD(4) EIU 3 6 words
LSN CMD(6)	LSN CMD(6)	LSN CMD(6)	LSN CMD(6)	LSN CMD(6)	LSN CMD(6)	LSN CMD(6)	LSN CMD(6)
FF1 RTN * WD 1 word	FF2 RTN * WD 1 word	FF3 RTN * WD 1 word	FF4 RTN * WD 1 word	FA1 RTN * WD 1 word	FA2 RTN * WD 1 word	FA3 RTN * WD 1 word	FA3 RTN * WD 1 word
ADTA 1 6 words	ADTA 2 6 words	ADTA 3 6 words	ADTA 4 6 words	Delay	HYD SYS 3 pressure C CD14 CH28 1 word	OMS left engine chamber press CD6 CH18 1 word	OMS right engine chamber press CD6 CH18 1 word
FF1 FROM Seq 2-6 36 words	FF2 FROM Seq 2-6 36 words	FF3 FROM Seq 2-6 36 words	FF4 FROM Seq 2-6 36 words				
FF1 IOM BITE 4 words	FF2 IOM BITE 4 words	FF3 IOM BITE 4 words	FF4 IOM BITE 4 words	FA1 PROM Seq 3-10 54 words	FA2 PROM Seq 3-10 54 words	FA3 Prom Seq 3-10 54 words	FA4 Prom Seq 3-10 54 words

LSN CMD = Listen command from PASS with WIX index noted in parentheses.

14.3.2.3 Max Rate Transfer Data Processing

The PASS 25 Hz 1A redundant set transfer consists of 12 half words containing such information as BCE element bypass status, port mode status, and major/ minor cycle status (table 14-III). Barring PASS failures, four identical copies are transmitted to the BFS via buses FC5-FC8 (e.g., copy 1 is transmitted on FC 5 data bus and is identified with string 1). The BFS compares (bit-by-bit) all copies of BCE element bypass status received and only establishes tracking on those strings (two or more) that agree, and downmodes those strings that disagree with the majority opinion. The source for the remaining PASS transfer data is the copy corresponding to the lowest numbered string being tracked by the BFS (such data are not compared with the other copies of the transfer data). Note that if a two-on-two split occurs, the BFS remains tracking the string pair containing string 1.

The BFS then compares (bit-by-bit) the resultant PASS version of the BCE element bypass status with the BFS's own version of the bypass status. Should the PASS and BFS disagree, even on a single element, the BFS will downmode (on first detected occurrence) the affected string. The intent of this software mechanization is to minimize the risk of BFS pollution due to bad input data on the affected string. For similar reasons, the BFS is designed to avoid tracking a single string.

14.3.3 I/O RESET Processing

Although the BFS is designed to "DK listen" to PASS keyboard inputs, the BFS will "bit bucket" an I/O RESET EXEC message to a PASS controlled CRT. However, the BFS does indirectly recognize the crew request, but in a manner that at first may appear convoluted. Whenever a BCE element is bypassed by the BFS and PASS (otherwise, BFS will downmode the string), the BFS is designed to monitor the PASS 25 Hz transfer data for a change of state in the BCE element bypass status. If the BFS detects that the PASS has upmoded any BCE element(s) (i.e., element was bypassed in previous minor cycle's version of PASS BCE element bypass status but is not bypassed in current minor cycle's version of PASS bypass status), then the BFS will also upmode the same BCE element(s). Note that if an upmoded BCE element still results in an I/O error condition (e.g., failed or powered off serial I/O LRU), then both PASS and BFS will again concurrently but independently bypass the element.

On the surface, it would seem more straight forward for the BFS to simply honor the PASS keyboard I/O RESET EXEC entry. Unfortunately, the minor cycle during which the PASS upmodes a bypassed BCE element can vary to some extent (specific CRT selected for inputs and/or PASS CRT I/O error processing are some of the variables that result in such BCE element upmoding uncertainties). Consequently, BFS programming difficulties exist in attempting to precisely synchronize the upmoding of BCE elements between the PASS and BFS when using the keyboard inputs; that is, potential for downmoding BFS to standalone due to PASS/BFS BCE element bypass miscompares (as logic currently is implemented) or potential BFS pollution if such compare

logic is relaxed during element upmoding process. Hence, the approach to monitor the PASS transfer data to detect BCE element upmoding.

When a BFS keyboard I/O RESET is performed, the software determines if the BFS is engaged, standalone, or tracking PASS. The software then branches to the appropriate I/O reset software.

BFS Engaged - The software initiates the I/O reset function on the first minor cycle within the major cycle that does not have Alt Rate data inputs; i.e., minor cycle 2. One string is upmoded per minor cycle starting with string 1 and continues through string 4. Following the upmoding of string 4, on the next minor cycle, the payload buses are upmoded.

BFS Standalone - BOS attempts to sync and track the PASS. If tracking two or more strings, BOS initializes the element bypasses and bus port mode to the PASS 25 Hz 1A transfer data.

BFS Tracking PASS - The software determines if BFS is in the RUN mode; i.e., the GPC MODE sw is in RUN. Then a test is made to determine if all strings are being tracked by BFS (PASS and BFS have matching string port mode), if so, only the PL buses will be port moded to match the PASS transfer data PL buses status. If BFS is not tracking all strings, then BOS attempts to sync and track all PASS strings according to PASS element bypass status and string/PL port mode status in transfer data.

TABLE 14-III.- 25 HZ TRANSFER DATA (1A INPUT)

Half Word	Bits*	BTU	Description/BCE Bypass Element	Input I/O Transaction
0	0-3	FF 1-4	FF input PROM (BCE STRG A)	MFE
	4-6	FF 1-3	TACAN/RA	MFE
	12-14	FF 1-3	IMU	MFE
1	12-15	FA 1-4	FA input PROM (BCE STRG C) (6/MC ANLG/DISC)	MFE
2	0-3	FF 1-4	MDM return word	HFE
	4-7	FF 1-4	FF input PROM (BCE STRG B)	HFE
	8-11	FF 1-4	ADTA	HFE
	12-15	FA 1-4	MDM return word	HFE
3	0-3	FA 1-4	FA input PROM (BCE STRG D)	HFE
	4	EIU 1	EIU 1 port 1 (FC 5)	HFE
	5		EIU 1 port 4 (FC 8)	HFE
	6	EIU 2	EIU 2 port 1 (FC 6)	HFE
	7		EIU 2 port 4 (FC 8)	HFE
	8	EIU 3	EIU 3 port 1 (FC 7)	HFE
	9		EIU 3 port 4 (FC 8)	HFE
	13	FA2	Hydraulic SYS 3 pressure C	HFE
	14	FA3	OMS left engine PC	HFE
	15	FA4	OMS right engine PC	HFE
4	0-2	FF 1-3	MTU	MTU
	4-5	FF 1, 3	NSP discretes	NSP
	6-7	FF 1, 3	NSP serial I/O data	NSP
5	-	-	BCE element bypass**	
6	-	-	MDM port mode status	
7	-	-	Major/minor cycle index	
8	-	-	Mission phase sequencing	
9	-	-	Spare	
10	-	-	Spare	
11	-	-	Spare	

*Unspecified bits in each BCE element bypass word have indeterminate values.
**Not utilized by BFS.

TABLE 14-III.- Concluded

WORD 6 - Data bus mode indicator

Bit 0	Unused
1	String 1 (FC 1, FC 5)
2	String 2 (FC 2, FC 6)
3	String 3 (FC 3, FC 7)
4	String 4 (FC 4, FC 8)
5	Payload

Set by PASS to indicate FC and PL buses are moded to the following ports:

Bit state	Strings 1-4	Payload	
		PF 1	PF 2
1	Primary	1	2
0	Secondary	2	1

WORD 7 - Major/minor cycle index

Bits 0-8 = (Indeterminate)

Bits 9-10 = Major cycle index

Bits 11-15 = Minor cycle index

The minor cycle index value increments from 0 (major cycle start through 23 and is referenced to the PASS/BFS shared inputs.

The major cycle index increments from 0 through 3 and is referenced to have PASS NAV quiescent at minor cycle index 12 of major cycle index 0 (Q-point).

WORD 8 - Mission phase sequencing

16 bit integer indicating current major mode of PASS

Bit = 1 indicates the specified element of an I/O chain has been bypassed by PASS.
Bit = 0 indicates that the specified element of an I/O chain has not been bypassed by PASS.

14.3.4 Engage/Disengage Transition

Once every minor cycle immediately prior to processing the Max Rate jobs, the engage/terminate B discretes are monitored for change of state. Table 14-IV presents the discrete configurations and software response.

TABLE 14-IV.- ENGAGE/DISENGAGE RESPONSES

Engage 1, 2, 3 (DI 35, 36, 37)	Terminate B (DI 13)	Software response
2 of 3 = 1	0	Engage
3 of 3 = 1	0	Engage
3 of 3 = 0	1	Disengage
Any other combination		Log "Illegal Engage/ Terminate B" Error Code(41 HEX) in GPC Error Log

14.3.5 Startup (HALT-STANDBY)

Following IPL and the repositioning of the GPC MODE switch to STANDBY, the BSL loads the BFS software when the ITEM 2 EXEC is performed on the IPL MENU display. The transition from the BSL to BOS occurs when the IPL source switch is repositioned to OFF. BOS post IPL initialization unprotects and zeros the BSL program, reads the IOP discrete words A/B, and determines the BFS GPC ID (DI 32, 33, 34). The GPC ID is used by the uplink software to accept/reject uplink loads.

Next BOS processes the system reset interrupt (previously set when moding to STANDBY). This interrupt handler software determines if a HISAM dump is being requested, sums protected memory and computes a checksum value, initializes MET = 160 ms and GMT to 1 day, zeros I/O compools, zeros GPC and I/O Error Logs, forces OPS 0, and issues an IOP master reset interrupt. Following IOP initialization, BOS attempts syncing and tracking PASS, performs engage transition if legal engage/terminate B discretes exist, or remains standalone or tracking PASS. Note that the FC and PL data bus port mode configuration will match the PASS 1A transfer data port mode indicators. Note that the BOS post IPL initialization of zeroing BSL, determining BFS GPC ID, etc. is not repeated when the GPC has previously been BFS IPL'd. Instead, BOS immediately starts processing the system reset interrupt following the mode transition from HALT to STANDBY.

14.3.6 Protected Memory Checksumming

BFS GPC memory is composed of protected and unprotected 16-bit halfwords. Each halfword is accompanied by a parity bit and a protect bit. The BOS at startup (OPS 0 initialization) sums all protected halfwords to establish a checksum value. Also, approximately every 16 seconds (depends on background processing time available), all of protected memory is resummed and verified in background processing. In order to reduce background checksumming time, the startup checksum software builds a table in each memory sector, identifying 16 halfword blocks of continuous protected halfwords. Each bit in the table represents 16 halfwords of memory. If the bit is set, then all 16 halfwords are protected. If the bit is not set, then the 16 halfwords are mixed protected and unprotected, or all unprotected. All of protected memory is still summed in background checksumming, the time is saved by not having to test each halfword protect bit within the 16 halfword block when the bit for that block is set (all protected).

The checksum software actually sums memory in increments of 512 halfwords until all of protected memory is summed. When a checksum miscompare is found, the "GPC SUM" fault message is annunciated. The fault message is suppressed following a G-Memory Write (from BFS keyboard), a Two-Stage-Command Write G-MEM Scatter or Write G-MEM Contiguous uplink command, or a change in the number of PASS strings being tracked (unprotect, update, reprotect strings tracking WIX table). Whether protected memory is altered or not, the checksum value is recalculated and the sector protect table regenerated.

During the process of summing protected memory, parity is verified for both protected and unprotected memory locations. If a parity violation is detected, a GPC parity error is logged and either a restart is commanded if in first pass BOS OPS 0 initialization, or the SUMCK module quits if in cyclic background summing.

Protection has been added to protect against restarts. The checksumming software is reinitiated when a restart occurs.

14.3.7 Application Transfer Data Inputs

In addition to the 25 Hz 1A transfer data inputs used by BOS to initialize the I/O data bus function, BOS receives 1.04 Hz 1B transfer data inputs for OPS 1 and 3 initialization and also G9 one shot inputs during prelaunch.

14.3.7.1 1.04 Hz Transfer Data (OPS 1 and 3)

The GN&C application software requires initialization data to perform backup flight critical functions. These initialization data are transmitted on FC 5, 6, 7, 8 buses in identical copies. The lowest numbered copy that BFS receives without I/O error is used by GN&C applications. Table 14-V identifies the transfer data contents.

TABLE 14-V.- PASS TO BFS 1.04 HZ TRANSFER DATA

Item No.	FSSR symbol	Coord. frame	Units*	Description
1	T_STATE	NA	SEC	State vector time tag
2	R_AVGG	M50	FT	Position vector
3	V_AVGG	M50	FPS	Velocity vector
4	V_ME_OUT	M50	FPS	V_REL_MAG at main engine failure
5	TBO	NA	SEC	Time base zero
6	Spare			22 words
7	Q_SM_M50	NA	NONE	M50 to stable member reference
8	Spare			24 words

*The units in the ICD override the units in the FSSR.

14.3.7.2 - G9 One Shot Transfer Data

The GN&C application software also requires calibration data generated prelaunch in G9 PASS functions. Approximately 30 minutes before launch, while the BFS is in OPS 0 and PASS is in G9, the G9 one shot data transfer is performed. The reception of the inputs must be enabled in the BFS (GPC Memory SPEC - Item 25) and then commanded in the PASS (via DEU equivalent uplink). The BFS receives four identical copies (on FC 5, 6, 7, 8 buses), verifies no I/O errors, and verifies checksum. The application uses the lowest numbered copy that has no errors. Table 14-VI identifies the transfer data contents.

TABLE 14-VI.- PASS GNC 9/GFS DATA TRANSFER - ONE SHOT

Item No.	FSSR symbol	Coord. frame	Units*	Description
1	SUMWORD	N/A	None	Sum of all halfwords in transaction with overflow ignored
2	PPOLYI	N/A	Ft/sec	Segment start values for independent variable of pitch and yaw polynomial relative velocity
3	THETIL	N/A	None	Segment start values for dependent variable of pitch polynomial tan pitch ang/4
4	PSII	N/A	None	Segment start values for dependent variable of yaw polynomial tan yaw ang/4
5	PHII	N/A	None	Segment start values for dependent variable of roll polynomial tan roll ang/4
6	GYREST DFX DFY DFZ	Platform	Rad/sec	IMU gyro restraint drift
7	GYMSPIN DSX DSZ	Platform	Deg/hr/g	Gyro spin axis mass unbalance drift
8	GYMOUT DOZ	Platform	Deg/hr/g	2-gyro output axis mass unbalance drift
9	ACCSFLO KIX KIY KIZ	Accel	μg/g	Low gain accel scale factor

TABLE 14-VI.- Concluded

Item No.	FSSR symbol	Coord. frame	Units*	Description
11	V97U4153C		FP psia	Chamber compensation bias
	V97U4154C		FP psia/bit	Chamber coefficient
	V97U4156C		FP psia	Chamber compensation bias
	V97U4157C		FP psia/bit	Chamber coefficient
	V97U4159C		FP psia	Chamber compensation bias
	V97U4160C		FP psia/bit	Chamber coefficient
	V97U4162C		FP psia	Chamber compensation bias
	V97U4163C		FP psia/bit	Chamber coefficient
	V97U4165C		FP psia	Chamber compensation bias
	V97U4166C		FP psia/bit	Chamber coefficient
	V97U4168C		FP psia	Chamber compensation bias
	V97U4169C		FP psia/bit	Chamber coefficient
12	V97U4429C		32 bit FLTG P.T, slugs	Estimated mass of orbiter with ET

*The units in the ICD override the units in the FSSR.

NOTE: The data associated with each listed item number (1-12) are transferred as a separate BCE element.

14.4 ERROR PROCESSING

In support of BOS controlling the I/O function and in dispatching jobs in an orderly manner, BOS monitors for/responds to error conditions. BOS monitors for I/O errors and eventually bypasses the failed input element. BOS responds to GPC errors by performing a software restart, continuing or redispatching the job.

14.4.1 I/O Error Processing

When BFS is in the engaged state, BOS is in full control of initiating input and output transactions. The MSC timer is loaded with a time large enough to allow completion of the entire input transaction (may contain several input elements per bus). For each input element, the BCE program is supplied with an initial T/O time which expires immediately following the input element completion. If either the MSC timer or BCE initial T/O timer expires without successful completion of the input transaction, an error condition exists. Also, an error condition exists if a hardware-detected error condition exists; e.g., invalid Manchester code, parity error, etc. is detected. For all cases, the MSC detects that an error has occurred and notifies BOS via an External 2 interrupt.

BOS responds to the External 2 interrupt by determining which data bus had an error and invokes the appropriate data bus error handler; i.e., DK, LDB, IP, PL, or FC bus. All error handlers initially operate in the same manner; i.e., logging the error in the I/O Error Log; however, bypassable and non-bypassable elements are treated differently. For bypassable elements, each I/O error is logged, the cumulative error counter incremented, and if a second consecutive error occurs, the second error is logged, error counter incremented, the element bypassed, and "I/O ERROR XX" fault message annunciated. For non-bypassable elements (e.g., DK, LDB, IP inputs), two I/O errors are logged, error counter incremented, and fault message annunciated. DK input error fault message I/O ERROR XX is annunciated after the first occurrence while IP input error fault message is annunciated after the second consecutive error. Additional errors will not be logged until after one error-free input. Logging of only one and/or two error(s) keeps the I/O Error Log from being flooded by non-bypassable errors.

The error processing while in the tracking PASS mode operates essentially the same as above except the BCE initial T/O timers are loaded and started following receipt of the LSN CMD and decoding of the BCE index. The BCE index informs BOS as to which BCE program to start for receiving the upcoming input data.

The I/O Error Log has two BFS unique items; i.e., element number and major/minor cycle. As originally designed, BOS logged the memory address related to the input element experiencing the error; however, every time the code was recompiled, the address would change, the SCP's would have to be redone. Somewhere around STS-2, the address was replaced with element numbers which specifically identify input elements. Table 14-VII provides the cross-reference between element number and input element experiencing the error. Lastly, BOS designers have never had a requirement to time tag errors other than identifying the major/minor cycle in which the error was detected. The fault message time can give an approximate time when the error occurred.

Output errors are detected by the BCE; i.e., the MIA is busy when time to perform an output transaction. BOS logs the appropriate I/O error data in the I/O Error Log.

14.4.2 GPC Error Processing

GPC error processing consists of responding to hardware- and software-detected errors. The response is to log the appropriate error in the GPC Error Log and to continue or restart the function being performed when the error was detected. BOS is notified of hardware-detected errors by interrupts; i.e., the hardware logic detecting the error issues an interrupt which BOS immediately responds to with interrupt-unique response software. Software-detected errors are also serviced immediately since the error detecting software branches to the error response software.

TABLE 14-VII.- BCE ELEMENT NUMBERS

EL no.	BCE element
00	EIU1
01	EIU2
02	EIU3
03	SPARE
04	FA PROM 24/MC
05	FF PROM 24/MC
06	ADTA
07	OMS/HYD
08	FF PROM 6B/MC
09	TACAN/RA
0A	MSBLS (1)
0B	IMU
0C	SPARE
0D	STAR TRACKER (1)
0E	REND RADAR (1)
0F	FA PROM 6B/MC
10	MTU
11-12	NOT USED
13	NSP DISCRETES
14	NSP DATA
15	NOT USED
16	FF RETURN WORD
17	FA RETURN WORD
18-1E	SPARE
1F	XFER DATA
20	PF1
21	PF2
22	FLEX1
23	FLEX2
24-25	NOT USED
26-27	SPARE
28	24/MC LSN CMD FC5-8 (2) P
29	24/MC LSN CMD FC1-4 (3) R
2A	SPARE I
2B	6A/MC LSN CMD FC1-4 (4) M
2C	6B/MC LSN CMD FC5-8 (5)
2D	6B/MC LSN CMD FC1-4 (6) P
2E	SPARE O
2F	1B/MC LSN CMD FC1-4 (7) R
30	XFER LSN CMD FC5-8 (8) T
31	SPARE
32	24/MC LSN CMD FC5-8 (2) S
33	24/MC LSN CMD FC1-4 (3) C
34	6A/MC LSN CMD FC5-8 (4) N
35	SPARE D
36	6B/MC LSN CMD FC5-8 (5)
37	6B/MC LSN CMD FC1-4 (6) P
38	1B/MC LSN CMD FC5-8 (7) O
39	SPARE R
3A	XFER LSN CMD FC5-8 (8) T
3B	SPARE
3C	ONE SHOT XFER
3D	ENGAGE INIT INPUTS
3E	LDB POLL
3F	DEU POLL
40	24/MC FC OUTPUT (HFE)
41	12/MC FC OUTPUT (DDU)
42	6B/MC FC OUTPUT (SPI)
43	6B/MC PYLB OUTPUT
44	UPLINK RTC OUTPUT
45-4F	SPARE
50	PCM (NON OI)
51	PCM OI - OA02
52	PCM OI - OA03
53	PCM OI - OA01
54	PCM OI - OF01
55	PCM OI - OF02
56	PCM OI - OF03
57	PCM OI - OF04
58	PCM OI - DC02

NOTES:

(1) These elements are not currently supported by BFS.
(2) HFE inputs LSN CMD FC5-8
(3) HFE inputs LSN CMD FC1-4
(4) NSP inputs LSN CMD FC1-4
(5) MFE inputs LSN CMD FC5-8
(6) MFE inputs LSN CMD FC1-4
(7) MTU inputs LSN CMD FC1-4
(8) XFER DATA inputs LSN CMD FC5-8

Hardware Errors - The IOP/CPU logic is continually monitoring for parity bit violations, store protect violations, or whether the number crunching is becoming too large or small to be accurately represented in a register. The response varies from no response other than logging the error in the GPC Error Log and continuing execution, to the most severe response of reinitializing the IOP electronics and restarting the software execution. Table 14-VIII presents the error codes logged for each type of error, and the recovery. A power transient causes the PRX interrupt code to be logged, a restart to be initiated, and "GPC PWR" fault message to be annunciated.

The watchdog timer is reset to zero once per major cycle and overflows at 3.14 seconds. When BOS is executing software nominally, the timer should not become greater than .96 seconds and only overflows (greater than 3.14 seconds) when BOS cannot continue to execute through major cycles. Upon overflowing, an External 0 interrupt is issued with bit 0 set. BOS attempts to log the GPC error code F0 in the GPC Error Log; however, software execution is either stopped or in a tight loop. In any event, the watchdog timer overflow (MSB) condition turns on the GPC I-Fail CAM light, and the CRT's will probably have big X's and POLL FAILs.

Software Detected Errors - Interrupts are not used to notify BOS of software-detected errors. Instead, the software detecting the error also logs the appropriate error code data into the GPC Error Log, and then branches accordingly; i.e., into restart software or ignore and continue. BOS and application software have tests for off nominal operations; e.g., the navigation routine attempts to integrate a bad state vector and uses all the allotted CPU processing time (processing overload). Also, error codes are logged in the GPC Error Log for PASS tracking difficulties when strings are downmoded. These errors are usually accompanied by I/O errors also being logged in the I/O Error Log.

14.4.2.1 Restart

Restarts can be commanded for both hardware- and software-detected errors. Irrespective of how the error is detected, if the error is caused by a hardware failure, then the IOP Master Reset commanded by the restart software will have a chance of clearing the failure. Hard failures will persist in causing continuous restarts, thus making the BFS no-go, the BFS CRT(s) will have big X's and POLL FAILs, and the BFS I-Fail CAM light will be illuminated (watchdog timer overflows 3.14 seconds).

The restart software is entered from either the error detection software (in BOS or applications) or from interrupt handler software. The appropriate error code is logged in the GPC Error Log and the fault message is cued for annunciation after the restart. The restart software halts the MSC and BCE programs and commands an IOP Master Reset to reset the IOP electronics. The restart software waits for the External 0 interrupt indicating that the IOP has been reset and is in control/monitor idle mode. The External 0 interrupt handler software performs IOP initialization, reenabling the appropriate MSC and BCE programs, etc., and then performs the tracking PASS function

TABLE 14-VIII.- BOS ERROR CODES AND RECOVERY

Error code	Interrupt type (a)	Error condition	Comment	Recovery (b)
01	EX1	CPU PCI data parity	CPU detected parity error during IOP response to PCI command word	RESTART
02	EX1	CPU DMA write parity	CPU detected parity error during IOP write of DMA data to CPU	RESTART
03	EX1	CPU DMA address specification	IOP attempted to write at a nonexistent memory location	RESTART
04	EX1	CPU DMA store protect violation	IOP attempted to write in store protected location	RESTART
05	EX1	CPU DMA address parity	During DMA via IOP, address received by CPU has bad parity	RESTART
06	EX1	AGE	Ground test only	CONTINUE
08	EX1	IOP PCI/PCO channel parity	Parity error detected during any of the following CPU/IOP transfers: Cmd word for PCI transfer, cmd word for PCO transfer, or cmd data word or PCO transfer	RESTART
09	EX1	IOP DMA instruction read parity	Parity error detected for a BCE or MSC instruction during DMA	RESTART
0A	EX1	IOP DMA data read parity	Parity error detected for a cmd word or cmd data word during DMA	RESTART

[a]The interrupt types are: PRX - Power Transient, MCK - Machine Check, PCK - Programmable Check, IMR - Instruction Monitor, EX0 - Level A Interrupt, EX1 - Level B Interrupt

[b]For ERROR TRAP refer to paragraph B.

TABLE 14-VIII.- Continued

Error code	Interrupt type (a)	Error condition	Comment	Recovery (b)
0C	EX1	IOP DMA queue overflow	DMA queue capacity of 64 requests exceeded. DMA request is lost	RESTART
0D	EX1	IOP DMA timeout	DMA transaction not completed within 8 μs	RESTART
0F	PRX	Power transient	IOP or CPU power supply detected an input voltage <17.5 V dc for >400 μs	RESTART
1X	EX2	Unidentified IOP programmable interrupt	The interrupt code retrieved from the level C Interrupt Register in the IOP contained an unidentified bit pattern	CONTINUE
20	EX3	Spare		CONTINUE
30	EX4	Spare		CONTINUE
40	SWR	Sum check fail	A 512-word block of memory failed sum check	CONTINUE
41	SWR	Illegal engage term B	A change in I/O Term B or Engage Discretes A, B, C resulted in an illegal combination	CONTINUE
43	SWR	Engaged in OPS 101	BFS FC transmitters are disabled in OPS 101	CONTINUE
44	SWR	String(s) downmode from 2 consecutive LSN errors	BFS failed to receive the HFE LSN cmd from the PASS two consecutive cycles	CONTINUE

[a]The interrupt types are:
PRX - Power Transient
MCK - Machine Check
PCK - Program Check
IMR - Instruction Monitor
EX0 - Level A Interrupt
EX1 - Level B Interrupt

[b]For ERROR TRAP refer to paragraph B.

TABLE 14-VIII.- Continued

Error code	Interrupt type (a)	Error condition	Comment	Recovery (b)
45	SWR	String(s) downmode from RS check	String(s) downmoded because of erroneous XFER data this cycle	CONTINUE
46	SWR	String(s) downmode from PFS/BFS XFER_ELEMENT_DISCREPANCY	String(s) downmoded because the PASS and BFS did not agree on the bypass status of elements this cycle	CONTINUE
47	SWR	String dowmode: last string	The BFS cannot track less than 2 strings; BFS goes STAND-ALONE	CONTINUE
81	MCK	IOP Memory address parity, accessed by CPU or IOP	Parity error detected on an address supplied by either the IOP or CPU	ERROR TRAP
82	MCK	CPU or IOP memory parity, accessed by IOP	Memory word content is not used	ERROR TRAP
83	MCK	CPU memory parity accessed by CPU	Memory word content is not used	ERROR TRAP
84	MCK	IOP memory parity, accessed by CPU	Memory word content is not used	ERROR TRAP
85	MCK	CPU ROS parity	Parity error detected in the read only storage	ERROR TRAP
90	PCK	Illegal instruction	A macroinstruction was encountered that had an illegal or unspecified op code	ERROR TRAP

[a]The interrupt types are: PRX - Power Transient, MCK - Machine Check, PCK - Program Check, IMR - Instruction Monitor, EX0 - Level A Interrupt, EX1 - Level B Interrupt

[b]For ERROR TRAP refer to paragraph B.

TABLE 14-VIII.- Continued

Error code	Interrupt type (a)	Error condition	Comment	Recovery (b)
91	PCK	Privileged instruction	An attempt was made to execute one of the following instructions while not in the supervisor state: ISPB, LPS, SSM, IC, PCI, PCO	ERROR TRAP
93	PCK	Address specification	CPU attempted to access a memory location >33FFF	ERROR TRAP
94	PCK	Fixed point overflow		ERROR
95	PCK	Significance	Zero raised to a power	CONTINUE
96	PCK	Divide inputs not normalized	The inputs to a floating-point divide were not normalized	ERROR TRAP*
97	PCK	Store protect violation	The CPU attempted to write into a store protected location	ERROR TRAP
99	PCK	Exponent underflow	---	CONTINUE
9A	PCK	Exponent overflow (convert)	---	ERROR TRAP*
9B	PCK	Exponent overflow Flt Pt	---	ERROR TRAP*
9C	PCK	Invalid Flt Pt divide	Divide by zero attempted	ERROR TRAP*
B0	SVC	Bad SVC	Undefined SVC type	CONTINUE
B1	SVC	HAL/S subroutine error	HAL subroutine encountered a programmer-defined error	CONTINUE

[a]The interrupt types are: PRX - Power Transient, MCK - Machine Check, PCK - Program, IMR - Instruction Monitor, EX0 - Level A Interrupt, EX1 - Level B Interrupt

[b]For ERROR TRAP refer to paragraph B.

TABLE 14-VIII.- Continued

Error code	Interrupt type (a)	Error condition	Comment	Recovery (b)
B2	SVC	Restart SVC[c]	Restart SVC has been issued	RESTART
C0	PC1	Max rate job overload	MSC is busy when it is time to start Max Rate Inputs or all the Max Rate jobs in the previous minor cycle have completed	RESTART
C1	PC1	Alt rate job overload	Alt Rate jobs still running at the time to begin Max Rate job preparation	RESTART
C2	PC1	Alt rate job overload	Alt Rate jobs continued into the next minor cycle	CYCLE EXTEND
E0	IMR	Instruction unprotected	An attempt was made to execute as an instruction an unprotected memory location	ERROR TRAP
F0	EXO	Watchdog timer	The watchdog timer has expired; the BFS has FAILED and will light his CAM light	CONTINUE
F1	EXO	IOP fail latch	This interrupt is associated with a FAILED CS GPC; therefore, this interrupt should not be seen	CONTINUE

[a]The interrupt types are:
PRX - Power Transient
MCK - Machine Check
PCK - Program Check
IMR - Instruction Monitor
EX0 - Level A Interrupt
EX1 - Level B Interrupt

[b]For ERROR TRAP refer to paragraph B.

[c]The RESTART SVC is the means by which jobs can request a restart where, in response to a BOS error, the BOS transfers control to a job error trap.

TABLE 14-VIII.- Concluded

Error code	Interrupt type (a)	Error condition	Comment	Recovery (b)
F2	EX0	IOP C/M idle	Any of the following occurred: IOP timing fault (IOP clock has failed) IOP ROS parity error CPU issued master reset PCO Machine reset issued by CPU power supply IOP HALT from GPC MODE switch IPL complete Master power on reset issued by IOP PS Voter fail latch OR timeout latch set and transmission termination control enabled Indicates that all hardware in IOP is RESET	RESTART
F3	EX0	IOP ROS parity	Causes all IOP hardware to be RESET	RESTART
F4	EX0	IOP timing fault	IOP clock has failed	RESTART

[a]The interrupt types are: PRX - Power Transient, MCK - Machine Check, PCK - Program Check, IMR -Instruction Monitor, EX0 - Level A Interrupt, EX1 - Level B Interrupt

[b]For ERROR TRAP refer to paragraph B.

if previously tracking PASS. If the BFS was previously tracking PASS, software execution will resume (BOS monitors PASS major/minor cycle) normal software execution at minor cycle 0 of the upcoming major cycle. If the restart occurred while the BFS was engaged or standalone, the software execution resumes at the next minor cycle. Note that software execution always resumes at a future minor cycle. For the engaged case, only one minor cycle's inputs/processing/outputs will be lost; and for the tracking PASS case, up to one major cycles's PASS command inputs and transfer data will be lost. For all restarts except those caused by a power transient, BOS keeps tally of the restart time and increments GMT and MET accordingly. Restarts caused by power transients open up the MTU tolerance to 10 seconds (only for the next MTU RM) and reinitializes GMT and MTU at the next MTU input cycle. Note that a "GPC PWR" fault message will be annunciated for power transient restarts and a "GPC BITE" message will be annunciated for all others.

14.4.2.2 Error Trap

Error trap was designed to handle those error conditions where the state of the processing and interrupt mask state have a bearing on the response. Figure 9-3 presents the logic steps and responses. Table 9-IX presents the responses for BSS jobs. Only dispatched jobs are affected and either will remain active if a cyclic job (NEXT) or quit if a one shot job (QUIT).

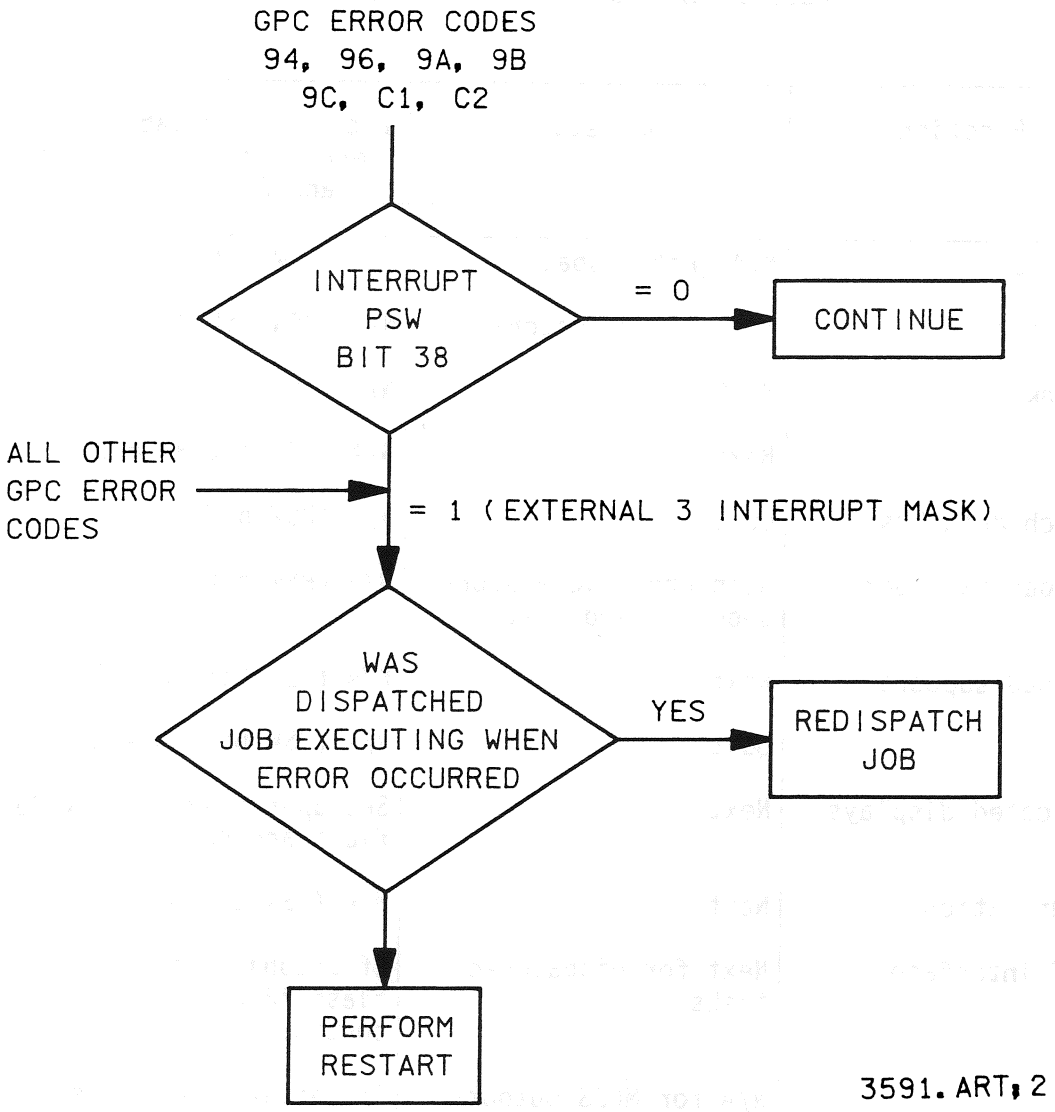


Figure 14-3.- Error Trap flow.

TABLE 14-IX.- BSS ERROR TRAP RESPONSE

Function	Job return	Recovery for table 1.24-I errors marked with "*" and PSW bit 38 = 0
BOS I/O	N/A (not dispatched)	N/A (PSW bit 38 = 1)
Downlist	N/A (not dispatched)	N/A (PSW bit 38 = 1)
Uplink	Next	N/A (PSW bit 38 = 1)
MTU RM	Next	N/A (PSW bit 38 = 1)
Launch data bus	Quit	N/A (PSW bit 38 = 1)
Payload bay doors	Stop Open/Close doors sequence and next	N/A (PSW bit 38 = 1)
Payload support	Next	N/A (PSW bit 38 = 1)
TFL	Quit	N/A (PSW bit 38 = 1)
Dedicated displays	Next	Set appropriate validity flags and continue
Annunciation	Next	N/A (PSW bit 38 = 1)
MCDS interface	Next for dispatched tasks	If graphic data, display flash "BAD DATA" and continue
	N/A for MCDS outputs (not dispatched)	If tabular data, display "????" and continue; if neither of above, ignore and continue

15 BFS INPUT/ OUTPUT

SECTION 16

BACKUP FLIGHT SYSTEM
SYSTEM MANAGEMENT AND SPECIAL PROCESSING

CONTENTS

Section		Page
16.1	OVERVIEW	16-1
16.2	GENERAL REQUIREMENTS	16-1
16.3	BFS SM FUNCTIONAL REQUIREMENTS	16-2
16.3.1	FDA	16-3
16.3.1.1	Limit Sensing	16-4
16.3.1.2	Fault Alarm Avoidance	16-5
16.3.1.3	Fault Annunciation	16-5
16.3.2	Scaling and Display	16-6
16.4	SPECIAL PROCESSING	16-7
16.4.1	SM Annunciated CRT Fault Messages	16-9
16.5	PAYLOAD PROCESSING	16-10
16.5.1	PM Functional Requirements	16-11
16.5.1.1	Fault Detection and Annunciation	16-11
16.5.1.2	Payload Downlist Processing	16-12
16.5.2	Payload Special Processing Functional Requirements	16-12
16.6	BFS SM DISPLAYS	16-13
16.6.1	SM SYS SUMM 1	16-14
16.6.2	SM SYS SUMM 2	16-15
16.6.3	SM THERMAL	16-16
16.6.4	SM PAYLOAD	16-17
16.7	BFS SM MONITORING DIFFERENCES BETWEEN OPS-0 AND OPS-1/3	16-17
16.8	BFS TMBU CAPABILITIES	16-18
16.9	ANNUNCIATION SYSTEMS	16-20
16.9.1	Types of Caution and Warning Functions	16-22
16.10	REFERENCES	16-28

SECTION 16
BACKUP FLIGHT SYSTEM SYSTEM MANAGEMENT AND SPECIAL PROCESSING

16.1 OVERVIEW

The Primary Avionics Software System (PASS) software contains the Guidance, Navigation and Control (GN&C) software whose primary function is to control the vehicle during all phases of flight. The PASS system, therefore, has very little capability to monitor other Orbiter systems. The Backup Flight System (BFS) was developed by Rockwell International/Downey (RI/D) to provide backup guidance, navigation and flight control functions, as well as primary systems monitoring and payload functions, during the ascent and entry phases of the Orbiter mission. During orbit operations, the system monitoring capability will be loaded into the System Management (SM) GPC.

The BFS SM function will provide information to the crew regarding the health, performance, and configuration status of the Orbiter subsystems and payload. Outputs from SM may result in commands to the payloads, alarm annunciation, CRT display data, dedicated display data and downlist data via systems software to the crew, to onboard recorders and to ground operations. SM also performs special computations and sequences to provide additional information to the crew and to provide assistance in the maintenance of vehicle subsystems.

16.2 GENERAL REQUIREMENTS

The BFS SM is divided into two sections called SM and Special Processing (SP). Both sections, which require the Backup Operating System (BFS Operating System (BOS)) support with respect to input data acquisition, CRT displays and outputs from the GPC to external hardware, are active during ascent and entry both preengage and postengage.

The SM/SP program is loaded into the BFS GPC with the BFS software. Activation of the BFS program shall also activate the BOS data acquisition for SM. The BOS will activate the SM/SP application controller at System Start and Restart times, and it runs cyclically once per major cycle. The SM/SP functions will be executed whenever the BFS GPC Mode switch is in either the STANDBY or RUN mode. A general diagram of the SM/SP and PL signal flow is shown in figure 16-1.

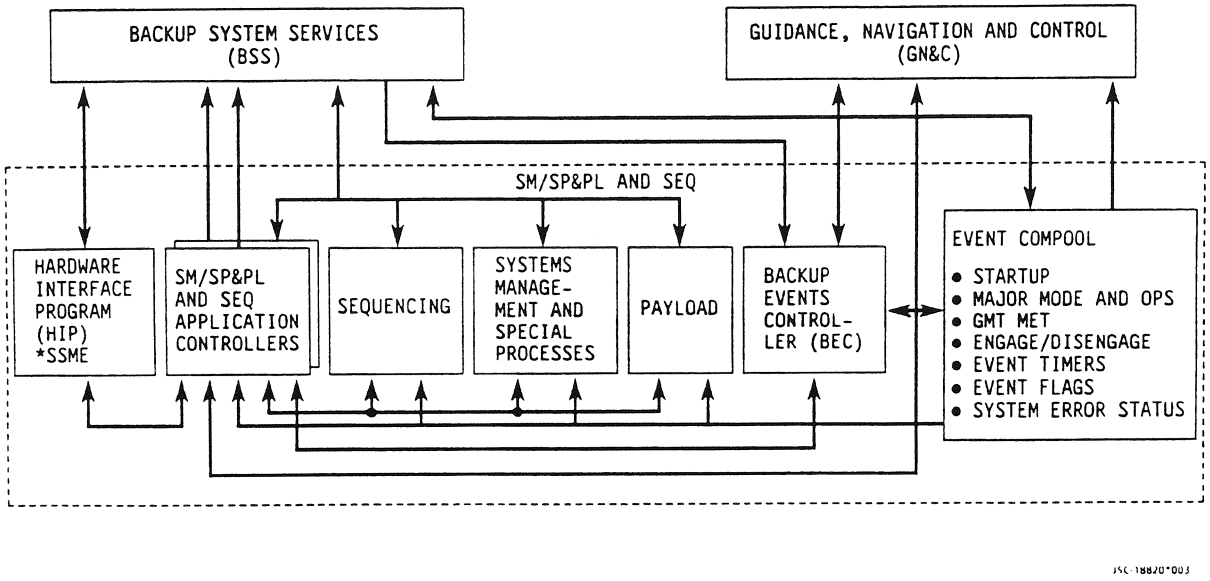


Figure 16-1.- SM/SP and PL General Signal Flow Diagram[1].

16.3 BFS SM FUNCTIONAL REQUIREMENTS[2]

SM is defined as the computer applications program that provides a backup capability to the Orbiter alarm subsystem. SM specifically does not include special processing (e.g., computations) and sequencing functions.

BFS SM consists of two principal functions: Fault Detection and Annunciation (FDA) and Scaling/Display (SD). The crew may, via the MCDS Interface software function, call up to three displays which will present a summary of the subsystem status on the BFS CRT. The SM displays, SM SYS SUMM1, SM SYS SUMM2, and THERMAL, are described later on in the brief. Some OI data parameters are processed by the FDA/SD functions during a limited on-orbit period only. Onorbit for BFS is defined as MM's 104 through 106 and 301 through 303, including a midphase called On-Orbit OPS-0. During these periods, the BFS FDA is supplemented with a software module termed ONORB FDASDT to perform SM FDA activities. BFS performs the SM class 2 and 3 annunciation during ascent major mode (MM) 104 through 106 to permit loading to SM OPS 2 software into a PFS GPC. During the BFS on-orbit OPS-0 phase (primary flight systems (PFS) OPS-2), BFS annunciates via CRT messages because PFS has assumed the SM FDA activities (e.g., PASS controls the 40-light C&W panel via its B/U C&W panel FDA software). During MM 301 to 303, BFS again performs expanded SM FDA annunciation activities since, as during the ascent phase, PFS (OPS-3) has no SM FDA capability. The FDA and SD functions are described below.

16.3.1 FDA

FDA (software module within the SM application system) is defined as that part of the SM application program that provides a backup capability to annunciate application and system software-detected faults via a common annunciation function to the Orbiter Caution and Warning (C&W) Subsystem.

FDA processing includes limit-sensing, false alarm avoidance and fault annunciation via BOS to the CRT, and external Avionics hardware for each FDA parameter which has not been identified to contain invalid data. The FDA function processes all FDA parameters in a cyclical fashion as acquired by the BOS Operational Instrumentation (OI) software. FDA processing is bypassed (no fault annunciation) for a parameter if input processing has detected invalid data for that parameter sample. All parameters containing invalid data are identified as such for each access.

The FDA function performs scaling, when required, on all data acquired by the FDA function in order that the parameter values and states may be displayed in engineering units (EUs).

The FDA function performed on several systems as follows:

A. Fuel Cell Stack Cool-Out Temperature

The three fuel cells stack cool-out temperature parameters have multiple limits. These limits are divided into two groups. Group 1 consists of the maximum and minimum temperatures allowable; Class 2 alarms are issued when these limits are exceeded. Group 2 consists of variable upper limits which depend on the fuel cells power outputs; Class 3 alarms are issued when these limits are exceeded. The fuel cell temperature as well as the out-of-limit indication are displayed on SYS SUMM 1 (fig. 16-2).

B. APU Turbine Speed

The three APU speeds shall have two upper alarm limits, depending upon whether or not the cockpit switch setting for APU Speed is high or normal. There is one lower limit. A class 2 alarm is issued for out-of-limit speed high (122 percent) and a class 3 alarm, for speed above (112 percent). The APU turbine speed low limit is 80 percent while the speed shutdown limit is 129 percent. The turbine speed as well as the out-of-limit indication are displayed on SYS SUMM 2 (fig. 16-3).

C. Gearbox Lube Oil Out Pressure Preconditioning

The three gearbox lube oil out pressure measurements are preconditioned so that the FDA alarm is inhibited when the APU is "off" (speed less than 40 percent). The oil pressures are displayed on SYS SUMM 2 (fig. 16-3).

D. PRSD FCP Reaction Valve Preconditioning

Each of the three reaction valve discretes for the SM SYS SUMM 1 display are derived from the O_2 and H_2 VLV positions. An open display indicator will reflect that both the O_2 and H_2 valves are open. A closed display indicator will reflect that either one or both valves are closed. The reaction valve positions are displayed on SYS SUMM 1 (fig. 16-2).

E. Brake Pressure Preconditioning

The 16 Landing Gear Brake Pressure measurements are preconditioned so that the FDA (alarm) is inhibited when the Landing Gear is down. The Landing Gear shall be regarded as down when both the Left and Right Main "Gear Uplocked" discrete measurements are "off." The scaled Brake Pressures shall be continuously displayed regardless of the Landing Gear position. The brake pressures are displayed on the THERMAL display (fig. 16-4).

16.3.1.1 Limit Sensing

SM has the capability to limit-sense analog parameter values, single-precision floating point parameters, and discrete parameters. The limit-sense test compares the parameter value with one set of two limit-set types: simple or multiple. A simple limit set for an analog parameter is one high-limit and one low-limit constant value. A multiple set is a set which contains two or three simple limit sets, from which one set is selected by preconditioning. Alarm Class 2 limit sets are simple limit sets only, whereas Alarm Class 0 or Alarm Class 3 limit sets may be simple or multiple. A description for the analog and discrete parameters are described below.

A. Analog parameters

The FDA function supplies the missing limit for analog parameters with only one limit defined. For a mission low limit, the FDA function supplies a low limit which is the value corresponding to the equivalent of the minimum PCM count. For a missing high limit, the FDA function supplies a high limit which is the value corresponding to the equivalent of the maximum PCM count. If the value of the analog parameter received is greater than the defined high limit or less than the defined low limit, the parameter is considered to be out-of-limit high or out-of-limit low, respectively. A down arrow shall be displayed for an analog parameter that is out-of-limit low and an up arrow for an analog parameter that is out-of-limit high. The analog-to-digital conversion convention is defined as follows:

TABLE 16-I.- ANALOG-TO-DIGITAL CONVERSION CONVENTION[2]

Volts	MSB LSB	Absolute decimal magnitude	Two's complement decimal magnitude
+5.11	V dc = 0 111 111 111 =	511	511 MAX positive value PCM count
+0.01	V dc = 0 000 000 001 =	1	1 MIN positive value PCM count
0.00	V dc = 0 000 000 000 =	0	0 Zero PCM count
-0.01	V dc = 1 111 111 111 =	1023	-1 MAX negative value PCM count
-5.12	V dc = 1 000 000 000 = Sign bit	512	-512 MIN negative value PCM count

B. Discrete parameters

For discrete parameters, only one limit is defined. If the state of the parameter agrees with the predefined state, then the parameter is considered to be within limit. When a discrete parameter is out-of-limit, the parameter status indicator (down arrow) is displayed adjacent to the parameter's state.

16.3.1.2 Fault Alarm Avoidance

The FDA function counts the number of consecutive FDA processing cycles for which the parameter value is out of limit. If a parameter has been out-of-limit for two consecutive cycles, fault annunciation for that parameter is enabled. A parameter fault annunciation so enabled, is not enabled again until the out-of-limit count has been reinitialized and the parameter has been again out-of-limit for two consecutive processing cycles. The out-of-limit count is reinitialized when the parameter value has been in limit for two consecutive processing cycles.

16.3.1.3 Fault Annunciation

The FDA function provides out-of-limit data to the BOS annunciation function to enable the annunciation of the detected fault. Warnings provided are Class 2 and Class 3, which are the C&W and Alert Systems, respectively.

The parameters that fail limit-checking are identified to the Backup Systems Services (BSS) MCDS Interface and Annunciation Function. Annunciation to the crew of detected failures is then accomplished via the Caution and Warning Panel, the alert tone and alert light, and the message line of the

CRT. The BFS SM function has three alarm classes and two indicators as follows:

Alarm Class:

Class	Alarm class
2	C&W
3	SM ALERT
0	Limit Sense/No Annunciation

Indicator:

Code	Alarm class
00	BACKUP C/W ALARM
15	PAYLOAD CAUTION

16.3.2 Scaling and Display[2]

All parameters processed by the FDA have their values and related status indicators displayed on the SM SUMM1, SYS SUMM2 and THERMAL displays.

A. Scaling

Dynamic scaling is provided for display of all analog parameters in EUs on the BFS displays. Scaling for each parameter is done by using one of the formulas shown below where each coefficient is real and single-valued and the "X" is the value to be scaled.

First Order $EU = A1(X) + A0$

Second Order $EU = A2(X)^2 + A1(X) + A0$

Third Order $EU = A3(X)^3 + A2(X)^2 + A1(X) + A0$

The A0, A1, A2, etc., are polynominal coefficients obtained from the function fitting to the calibration curves. For analog parameters whose PCM count is equal to or greater than +500 PCM counts, the parameters are considered OFF-SCALE HIGH. For analog parameters whose PCM count is less than or equal to zero PCM counts, the parameters are considered OFF-SCALE LOW. A parameter status indicator is enabled for display for both conditions. The BFS only has the requirement to provide third order scaling for a maximum of 10 parameters.

B. Parametric display

The BFS displays provide the current values/states and status indicators of all FDA parameters which are displayed in EUs with an update rate of once per major cycle. A single character space is reserved to the immediate right of each parameter value/state for the parameter status

indicator. When a parameter value has been determined to be invalid or missing, the last good value is displayed. When an analog parameter value has been determined to be off-scale high or off-scale low, the off-scale value is displayed. The legal parameter status indicator symbols and their meaning are shown below in order of priority (highest to lowest priority).

"M" - Invalid data or missing data as determined by input error processing.

"H" - Off-scale High. For analog parameters whose value received is equal to or greater than +500 PCM counts.

"L" - Off-scale Low. For analog parameters whose values received is equal to or less than zero PCM counts.

"↑" - Out-of-limits High. For analog parameters whose values have been determined to be out-of-limit high by the FDA function.

"↓" - Out-of-limit Low. For discrete parameters whose states have been determined to be out of limit by the FDA function. Also for analog parameters whose values are determined to be out-of-limit low by the FDA function.

16.4 SPECIAL PROCESSING[2]

Special Processing (SP) is defined to be application processing in the Backup Flight Computer which includes special computations and controlled processes. SP is not a standard or self-contained portion of the SM application programs. For example, FDA is a standard portion of the SM.

The BFS SP is performed on the following Orbiter subsystems:

A. Flash Evaporator/Ammonia Boiler Activation/Deactivation

The Active Thermal Control System (ATCS) contains two Freon loops which operate simultaneously throughout the mission. The Flash Evaporator and Ammonia Boiler provide cooling to the Freon loops during different portions of the mission. The Flash Evaporator requires a vacuum in which to operate while the Ammonia Boiler is activated during entry, and it remains ON during landing until approximately 15 minutes after rollout. The BFS commands the Flash Evaporator controller ON during ascent and turns the selected controller OFF during entry, and commands both Ammonia Boiler controller ON. Both the Flash Evaporator and the Ammonia Boiler are initialized to OFF in OPS-1. In OPS-3 or -6, the Flash Evaporator is initialized ON and the Ammonia Boiler is initialized OFF. In OPS-0 the Flash Evaporator is initialized OFF. Also in OPS-0 the Ammonia Boiler is initialized OFF, unless the OPS-0 transition is from MM 305 or MM 603, in which case it is initialized ON. The sequence will monitor various parameters, and issue commands at a process rate of once every even numbered major cycle.

B. APU Fuel Quantity

The sequence calculates the quantity of hydrazine fuel remaining in each of the three APU fuel tanks. This sequence is required for ascent and entry. The APU fuel is stored in a diaphragm-type tank with nitrogen used as the pressurant. The PVT method is used to determine the volume of nitrogen which in turn is subtracted from the volume of the fuel tank to arrive at the quantity of fuel remaining. Fuel tank temperature and pressure are the primary parameters for calculating APU fuel quantity. The output from the quantity calculation is displayed on a meter and in SYS SUMM 2 (fig. 16-3).

C. Fuel Cells Total Current and Power Computation

The sequence computes the total current and power of the three fuel cells during ascent and entry to provide a status of the Electrical Power Subsystem. The results of the computation, total current and power, can be made available for display on SYS SUMM 1 (fig. 16-2). The sequence will monitor the input parameters and also process the computed outputs at a rate of once every two major cycles.

D. Hydraulic Water Boiler Quantity

The sequence computes the amount of water remaining in the three storage tanks. The water is consumed in the spray boiler to cool the hydraulic fluid and APU lube oil. Water is stored in bellows type tanks, which are pressurized by gaseous nitrogen. The storage tanks supply water to the spray boiler as demanded by the hydraulic fluid and APU lube oil temperature. As the water is depleted, the percent amount of water remaining in each tank is computed and displayed on cockpit meter and in SYS SUMM 2 (fig. 16-3). The computed amount is also processed by FDA to alert the crew when a present minimum limit is exceeded.

E. S-Band Antenna Management

The sequence automatically selects antennas for S-band communication during ascent and entry. The BFS will select the optimum antenna for both the S-band Phase Modulation and Frequency Modulation (FM) communication. The antenna management function is performed at the rate of once every even numbered major cycle in all modes except OPS-0 in which automatic antenna management will be suspended upon transition to OPS-0.

F. Freon Pump Power Management

The sequence provides for automatic reconfiguration of the Freon pumps during a fuel cell failure. This capability is provided to implement the requirement that during ascent, in the event of a fuel cell failure, main bus power will not be immediately restored on the lost bus and critical loads will be transferred to other buses for the remainder of ascent.

G. Cabin Pressure Monitor

The sequence monitors the cabin pressure to determine cabin pressure change rate over a 30-second time period. The pressure change shall be compared to a predetermined leakage tolerance and displayed on SYS SUMM 1 (fig. 16-2). When the leakage tolerance is exceeded, a class 3 alarm and CRT message is issued. The sequence is scheduled for execution at transition to MM 101 and runs both pre- and post-engage. The process monitors the input parameter and performs the calculations at least once every five major cycles.

H. Fuel Cell End Cell Heater Monitor

Each fuel cell powerplant contains two end cell heaters which cycle on and off automatically to maintain a desired end cell operating temperature. Should any end cell heater cycle ON and remain ON for a period exceeding 6 minutes, crew intervention may be required to prevent a hazardous situation from occurring. This sequence monitors for that potential hazardous condition and, upon detection, alerts the crew via a class 3 alert tone and a SYS SUMM 2 indication (fig. 16-3).

I. Tire Pressure Monitor

Each of the 6 orbiter tires contain two pressure sensors. These sensors are connected to the Orbiter via "jerk out" wires which become disconnected upon release of the landing gear. The process monitors both sensors for each tire for FDA of a leak condition unless the "Landing Gear Down Arm Switch" is activated by the crew. Upon detection of a tire pressure leak condition, the crew is alerted via class 3 alert tone and the THERMAL display (fig. 16-4). The FDA lower limit can be updatable via a Table Maintenance Block Update (TMBU) function. The process runs and displays data in OPS 0, 1, 3 and 6. However, the FDA shall be performed in OPS 3 and 6 only.

16.4.1 SM Annunciated CRT Fault Messages[2]

The following fault messages are annunciated by the BFS SM function. The table below identifies the fault messages as well as the Alarm Class and the C&W Light indication. The Alarm Class 2 (C&W) and Alarm Class 3 (SM Alert) indications are both available on panel F7 in the forward station. The SM0, SM1 and SM2 indications on the Fault message represent the CRT display (THERMAL, SYS SUMM 1, SYS SUMM 2) which is used by the crew for system monitoring after a fault annunciation.

The fault messages for a Class 2 alarm which illuminate the BACKUP C&W light are shown below. The fault messages are grouped by CRT display.

SM1 CABIN N2
SM1 CABIN O2
SM1 CABIN FAN
SM1 CABIN HXT
SM1 CABIN PRESS
SM1 CABIN PPO2
SM1 FC PUMP 1,2,3
SM1 FC REAC 1,2,3
SM1 FC STACK T 1,2,3
SM1 MAIN BUS V A,B,C
SM1 AC OVLD 1,2,3
SM1 AC VOLTS 1,2,3

SM2 APU SPD HI 1,2,3
SM2 APU SPD LO 1,2,3
SM2 CRYO H2 PRES
SM2 CRYO O2 PRESS
SM2 CRYO O2 HTR
SM2 EVAP OUT T 1,2
SM2 FREON FLOW 1,2
SM2 H2O PUMP P 1,2
SM2 HYD PRESS 1,2,3
SM2 APU TEMP 1,2,3
SM2 AV BAY TEMP

The fault messages for a class 3 alarm which illuminates the SM ALERT light are shown below. The fault messages are grouped by CRT display.

SMO THRM APU
SMO THRM FRN
SMO THRM H2O
SMO THRM EVAP
SMO THRM HYD
SMO FC END HTR 1,2,3

SM1 CABIN IMU
SM1 CNTL/ESS V
SM1 FC STACK T 1,2,3
SM1 DC VOLT FC 1,2,3
SM1 FC AMPS 1,2,3
SM1 MAIN BUS V A,B,C
SM1 FC COOL P 1,2,3
SM1 FC EXIT T 1,2,3
SM1 EMERG dP/dT BU
SM1 FC PH 1,2,3
SM1 FC DELTA V 1,2,3

SM2 APU 1,2,3
SM2 AV BAY FAN
SM2 HYD QTY 1,2,3
SM2 HYD RSVR T 1,2,3
SM2 W/B QTY 1,2,3
SM2 APU SPD HI 1,2,3
SM2 HYD ACUM P 1,2,3
SM2 FU TK VLV APU

The Flight Data File (FDF) REFERENCE DATA (REF DATA) checklist contains the above class 2 and 3 alarm fault messages together with a quick reference to cause of alarm and any applicable comments.

16.5 PAYLOAD PROCESSING[2]

Payload requirements are divided into two sections. The first section is called Payload Management (PM) and encompasses FDA, Scaling, Display and Downlist processing for Shuttle Orbiter Payloads. The second section is called Payload Special Processes (PSP) and includes computations and controlled process not provided for under PM. Any parameter (measurement) required for payload processing is acquired via the BOS I/O interface once per major cycle. Payload processing is active during ascent and entry both pre and post-engaged.

The PM and PSP programs are loaded into the BFS GPC with the other BFS software. Both functions are enabled for all OPS and all Major Modes while the BFS GPC has control of the Payload buses. When the BFS GPC does not have control of the payload buses, the PSP may be invalid.

The Payload Support function provides downlisting of payload analog signals and discretes during ascent and entry phase. Payload command capability is provided by BFS processing of uplink commands through buses PF1 and PF2 during ascent and entry.

The Payload portion of SM does not include the Payload Bay Door (PBD) function. The BFS PBD program is one of the major functions performed by the BSS.

16.5.1 PM FUNCTIONAL REQUIREMENTS

The PM portion of the BFS consists of the table-driven processing which provide the capability for FDA, Scaling, Display and Downlist processing of up to 50 payload support parameters for any payload during ascent and entry. Payload processing will be disabled for all flights except those that require payload support from the BFS. When disabled, the payload display (SPEC 206) will not be available and no parameters are processed by the payload function.

The 50 Payload measurements which are processed by the BFS PM function are as follows:

Measurement number	Measurement name
V98U4901P	Payload Data Input Word 1
thru	thru
V98U4950P	Payload Data Input Word 50

16.5.1.1 Fault Detection and Annunciation

The PM FDA function, which includes Limit Sensing, False Alarm Avoidance, Fault Annunciation, and CRT display Interface for detected faults, is exactly the same as the FDA performed by the BFS SM function. The major difference is the Parametric Scaling of the analog parameters which is only done to the first order as shown in the formula below:

$$EU = A1(X) + A0$$

Appropriate Payload Display pages are available with the current values/ status indicators of all FDA parameters. The update rate of all parameter values/states and status indicators shall be once per major cycle. When a parameter value has been determined to be invalid or missing, the last good value will be displayed. For the payload data, there are only three status indicators, "M", "+" and "+", which have the same function as the status indicators mentioned in paragraph 16.3.1.4.B. The BFS alarm class for fault annunciation are the same as the alarm classes in the SM FDA function.

For the parameters displayed on the mission-dependent Payload display, there will be one fault message for class 2 alarm (206 PL CL 2) and one fault message for class 3 alarm (206 PL CL3).

The capability exists to output alarm annunciation data to the crew via the PF 1 or PF2 MDMs. If BFS is engaged, the capability exists to annunciate class 2 messages via the FF1 and FF3 MDMs. Indicator-dependent fault annunciation to the "GPC-Payload Caution" PL (or SM) indicator is only via PF2.

16.5.1.2 Payload Downlist Processing

There are only 25 slots available for downlisting parameters processed by the PSP functions. The 25 slots can be either 25 analog parameters, 50 discrete parameters, or a combination of both, ranging from 49 discretes and 1 analog, to 24 analogs and 1 to 16 discretes. Each slot can contain an analog or 1 to 16 discretes bits. The MSID parameters are as follows:

Measurement number	Measurement name
V98M0261P	Payload Downlist Word 1
|	|
thru	thru
|	|
V98M0285P	Payload Downlist Word 25

The MSIDs V98M0261P through V98M0264P are packed to include those parameters required in both Low (LDR) and High Data Rate (HDR). No provisions have been made for specifying LDR and HDR requirements for any of the Payload Special Processes function resultant parameters. Thus all Payload Special Processes function parameters shall be downlisted in the HDR only.

16.5.2 Payload Special Processing Functional Requirements

Payload Special Processes is defined to be application processing in the Backup Flight Computer which is not a standard or self-contained portion of the PM program. For example, Fault Annunciation is a standard portion of the PM function.

Those input parameters for the Payload Special Processes which originate in the specific payloads are considered to be reconfigurable from flight to flight and are defined by means of Payload Level C requirements. Any constants used in the Payload Special Processes are also considered to be reconfigurable and likewise defined on Payload "Level C Cards."

The Payload limit/states and constants can only be modifiable by the TMBU Uplink Function and by the Payload Preprocessor. All other payload tables are only modifiable by the Payload Preprocessor. The Payload limit/state tables and constant tables are contained in contiguous sections of memory for TMBU access.

16.6 BFS SM DISPLAYS[1]

The following displays are available at all times in the SM major function. The available displays and keyboard inputs necessary to invoke the display are as follows:

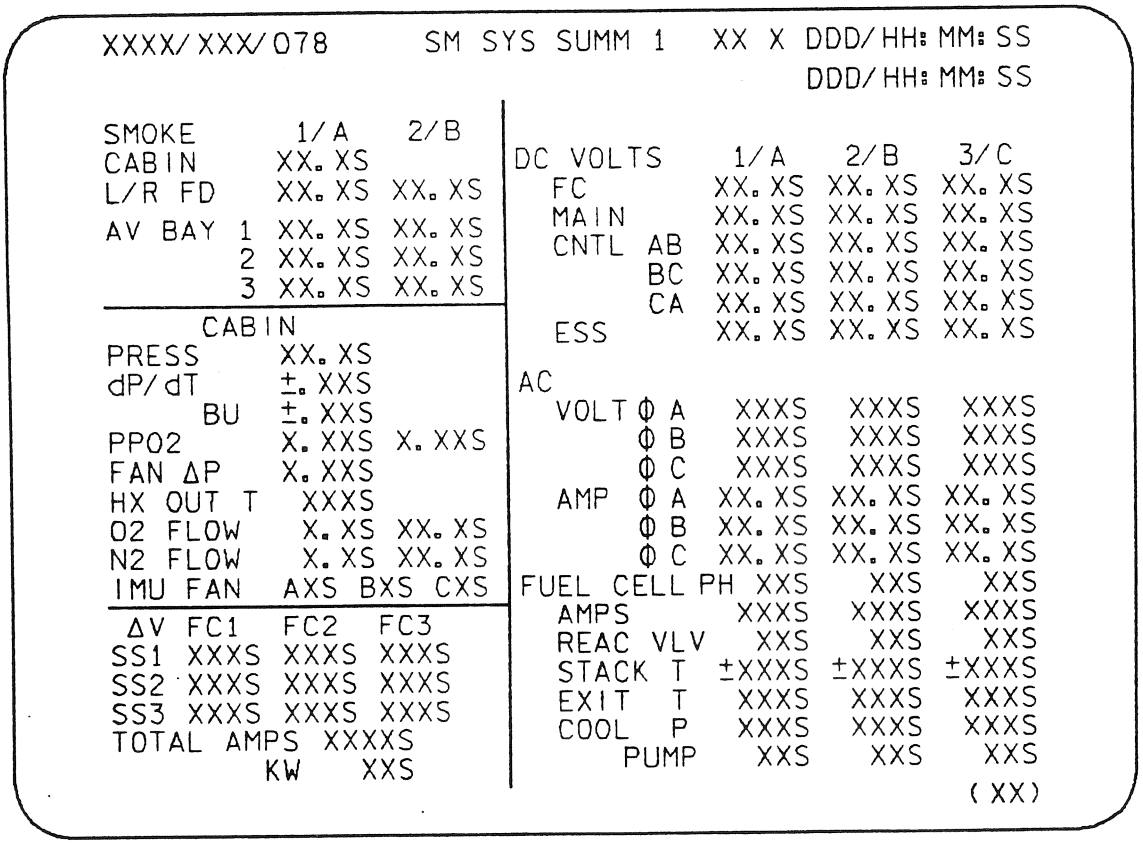
Display	Keystrokes
THERMAL	Major Mode Display/000
SM SYS SUMM 1	SPEC 78 PRO, SYS SUMM
SM SYS SUMM 2	SPEC 79 PRO, SYS SUMM
PAYLOAD	SPEC 206 PRO

The description of the CRT displays are shown below. Since there is only one OPS in the SM Major Function, the THERMAL/PL display is active at all times as the underlying OPS/Major Mode display.

The SYS SUMM 1 and 2 displays are the two primary formats for overall monitoring of the SM systems configuration and operational status. The display's parameters support the OI C&W function. The SYS SUMM 1 and 2 displays and all of its parameters are available to the BFS when the major function switch on panel C2 is in the SM position, by use of the SYS SUMM entry from the keyboard which alternately calls up the SM System Summary 1 and 2 display, or by a keyboard input of SPEC 78 PRO or SPEC 79 PRO, respectively. The THERMAL display is available to the BFS and is active at all times when the major function switch in panel C2 is in the SM position. In these displays, there are no display parameters that are dependent upon BFS engaged condition versus the nonengaged condition.

16.6.1 SM SYS SUMM 1 Display

The display presents parameters associated with the cabin atmosphere, avionics bay and IMU fans, electrical power distribution system (EPDS), and fuel cell parameters. There are practically no BFS/PFS display differences, except for the BFS computed BU dp/dt and fuel cell PH values.

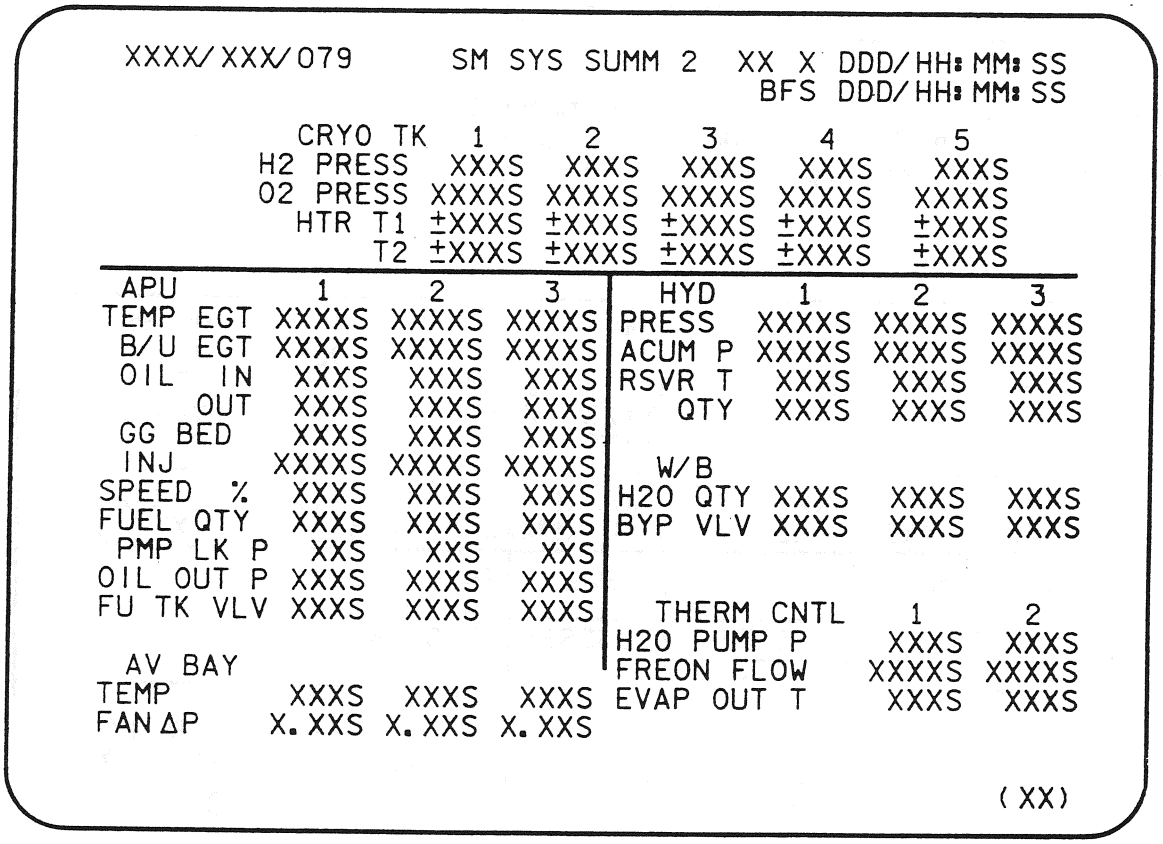


188201232.ART;2

Figure 16-2.- SM SYS SUMM 1 display.

16.6.2 SM SYS SUMM 2 Display

The display presents parameters associated with the fuel cell cryogenic tanks, APU system, avionic bays temperature, hydraulic system, water boiler and thermal controls. This display is unique to BFS during ascent and entry. On orbit, the PFS SM SYS SUMM display is used and is comparable to the BFS display with minor differences.



18820133.ART; 1

Figure 16-3.- SM SYS SUMM 2 display.

16.6.3 SM THERMAL Display

The display provides the following monitoring capabilities:

A. The temperature of the hydraulic fluid in the actuators and pressurization systems of the aerosurface control system.

B. The temperature in the propellant, flash evaporator, hydraulic water boiler and the APU.

C. The Freon loop and H_2O parameters necessary to support flash evaporator thermal monitoring.

The display is unique to the BFS. The majority of the parameters displayed are only valid in MM 104 to 106, MM 301 to 303, and on-orbit OPS 0. Only the hydraulic system temperatures and fuel cell end heaters are valid in OPS 0, 1, and 3.

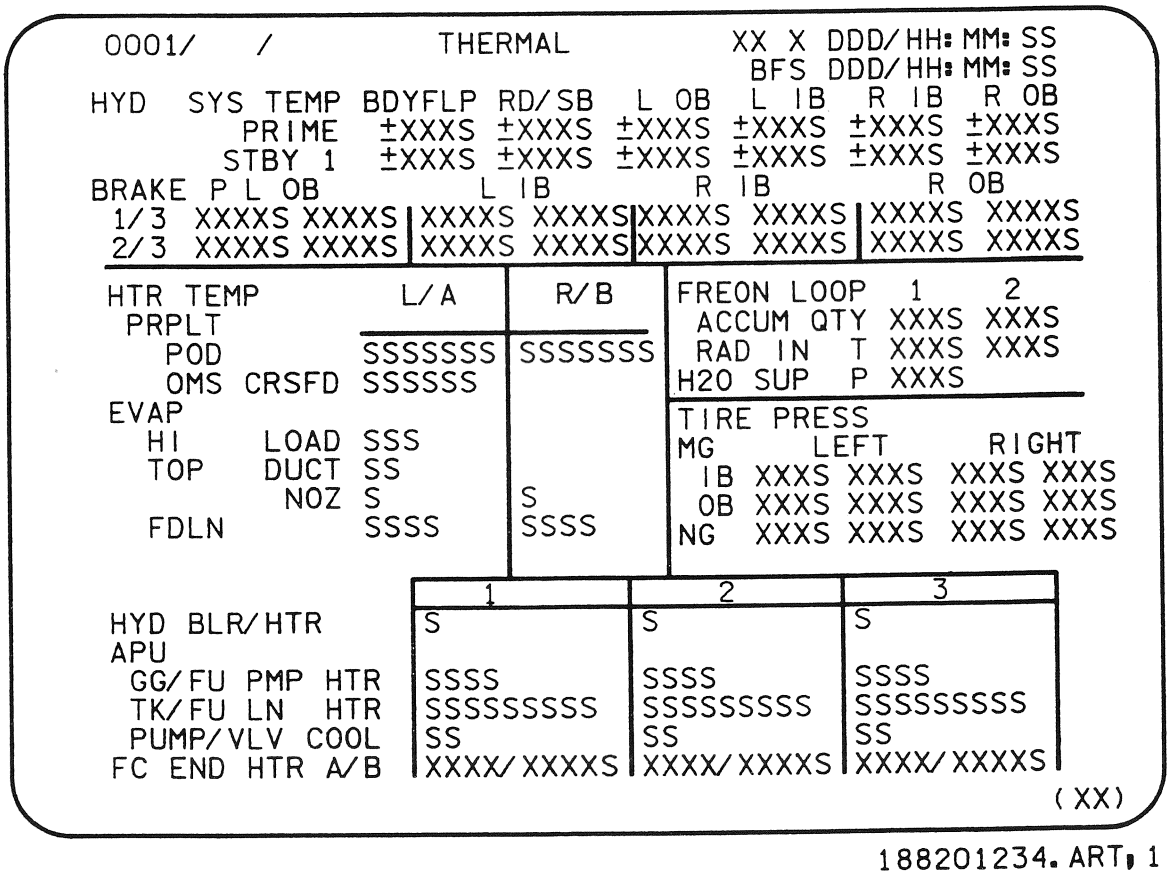


Figure 16-4.- THERMAL display.

16.6.4 SM Payload

The generic Payload Display SPEC 206 can be used to display up to 50 parameters in any combination of analog and discretes acquired from the MDMs and PCMMU at a rate of once per major cycle, except for the comm faulted data which maintains the displayed value at the time of the comm fault. These parameters and their mode of display are mission dependent. Only one PAYLOAD display will be available on any one flight.

The display, used as a Spacelab Launch/Entry display, is available only on flights requiring Spacelab support from the BFS. For this use, the display provides the following monitoring capabilities for the Spacelab:

A. The EPDS - display voltage, current and temperature values associated with the main and auxiliary buses, the subsystem and experiment inverters, and the status of each of the experiment power distribution boxes.

B. The module atmospheric pressure and the O_2 partial pressure within the module.

C. The delta pressure across the operating avionics fan.

D. The water pump inlet and outlet pressures on the H_2O loop and the delta pressure across the operating water pump.

The display is available to the BFS only and is active at all times when the CRT major function switch on panel C2 is in the SM position.

16.7 BFS SM MONITORING DIFFERENCES BETWEEN OPS-0 AND OPS-1/3[2]

There is only one OPS in the BFS SM Major Function. The SM function is therefore executed in all OPS whether its OPS 0 or OPS 1/3. The difference between OPS 1/3 and OPS 0 is strictly tied to which portion of the GN&C software is used. The BFS capabilities/functions while in OPS 0 are as follows:

A. BSS

1. Data bus configuration
2. Disengage PASS tracking initialization
3. GPC memory and fault summary displays
4. Uplink (if NSP inputs are available)
5. Downlist (ascent format)
6. One shot PASS data transfer (prior to MM 101 selection only via keyboard entry on GPC memory CRT displays)

B. GN&C

1. Application controller

C. SM/SP, SEQ and PL

1. Backup events controller
2. SM/SP, SEQ and PL application controllers
3. SM (except for antenna management)
4. Special processes
5. RCS Quantity gauging

Below is the table which contains the specific OPS/Major Modes in which the subsystem FDA is performed.

Name	Periodicity	OPS	Major modes
SM Initialization	On Demand		
FDA/SDT	1/MC	0,1,3,6	All
Annunciation	On Demand	0,1,3,6	All
Main bus voltage	1/MC	0,1,3,6	All
APU fuel gauging	1/MC	0,1,3,6	All
S-band antenna mgmt	1/2 MC	1,3,6	All
Hydraulic water boiler qty.	1/MC	0,1,3,6	All
Fuel cell total current and power gauging	1/MC	0,1,3,6	All
Flash Evaporator/ Ammonia Boiler	1/2 MC	1,3,6	All
Freon pump power mgmt	1/MC	1	All
On-orbit FDA/SDT	1/MC	0,1,3	104-106 301-303 On-orbit OPS-0
APU turbine speed	1/MC	0,1,3,6	All
Gearbox lube oil pressure	1/MC	0,1,3,6	All
Cabin press monitor	1/5 MC	0,1,3,6	All
Fuel cell end cell heater monitor	1/MC	0,1,3,6	All
SM brake pressure monitor	1/MC	0,1,3,6	All
Tire pressure monitor	1/MC	0,1,3,6	All

16.8 BFS TMBU CAPABILITIES

The SM controls selected functions onboard the vehicle and it provides the user capability to update parameters which control SM processing.

In the PASS SM OPS 2/4, control of the SM software limits may be changed via the SM TABLE MAINTENANCE SPEC 60 or via uplink. However, in the BFS, the software limits/states in the SM and PM function can only be modified by the

TMBU uplink function. The limits from the FDA tables are, for the most part, referenced by using the associated parameter MSID with "UL" or "LL" appended to the end for Upper Limits or Lower Limits, respectively. The limits are stored, as a table, in a continuous section of memory for access by the TMBU function.

Parameters, although previously I-loaded, whose precise values are considered to be undetermined prior to T-10 hours before launch, will have their values initialized via the TMBU function.

The following TMBUs are premission planned for STS-26 reflight as shown below. However, other TMBUs will be processed as the need arises for SM.

A. Mechanical

Rationale: This TMBU is required to correct the APU fuel quantity constant if KSC loads the APU fuel to an amount different from the OMRSD by a difference of ± 2 percent.

Retention: Mechanical is currently evaluating the need for this TMBU.

When uplinked: Prelaunch

B. Booster

Rationale: If an MPS LO_2 or LH_2 low level sensor fails prior to launch, that sensor is disabled to remove it from the SSME shutdown voting logic.

Retention: All flights.

When uplinked: Prelaunch, only after a sensor failure.

C. EECOM

Rationale: This TMBU will return freon coolant loop FDA to nominal on-orbit values

When uplinked: Post-MECO + 5 minutes.

D. PROP

Rationale: The day before launch, PROP will compare actual RCS loading quantity with the quantities that were loaded in the BFS. If the quantities are off by more than ± 3 percent, PROP will request the uplink of this TMBU prelaunch. Otherwise it will be sent post-MECO + 5 minutes.

Retention: All flights.

When uplinked: Post-MECO + 5 minutes (unless required to be uplinked prelaunch).

16.9 ANNUNCIATION SYSTEM[1]

The annunciation system provides aural and visual cues to apprise the crew of abnormal and hazardous conditions in the Orbiter operating systems that can cause immediate or impending danger to the crew and/or failure of critical Orbiter subsystem. The alarms classes which are applicable to the SM system are as follows:

1. Class 2: C&W - SM, Payload and GN&C C&W indicator alarm outputs and fault messages.

2. Class 3: Alert - alert light, alert tone output, and fault messages.

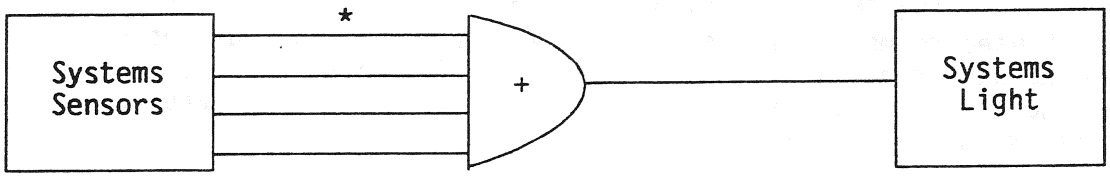
3. Class 0: Limit Sense - limit sensing that results in the output of a parameter status indicator with no associated fault message, alert tone, or light.

The annunciation system is comprised of the following elements:

A. The Primary C&W System

The Primary C&W System (not to be confused with the PFS) is a hardware system comprised of various transducers and sensor, C&W electronic unit (C&WEU), a 40-light C&W annunciator light panel on panel F7, and a C&W display and control panel (R13) which includes a 120-parameter status light array. A representative picture of the C&W light annunciation panel can be seen by each C&W type in the next paragraph. The C&WEU is a digital electronics which receives hardware inputs from various sensors throughout the Shuttle system and discretes from the BFS GPC. The unit performs limit tests on the various hardware inputs and utilizes the GPC software derived discrete inputs to provide various degrees of redundancy for the C&W Class 2 functions and to provide class 3 SM alert visual and aural indications.

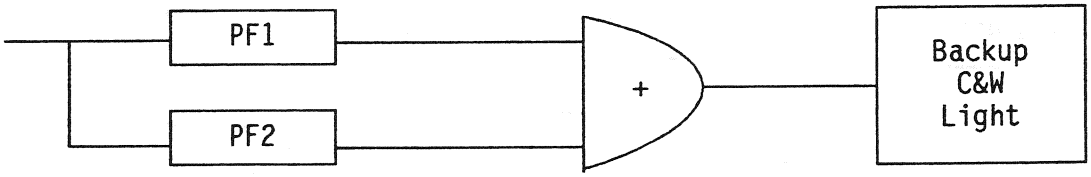
The C&WEU contains logic to illuminate the lights dependent on the function type. The function types dedicated to SM do not have a GPC Interface and therefore are unaffected by engage or mission modes. These types have a hardwired dedicated circuit which is used to drive a specific function-related indicator in the C&W annunciator light panel via the C&WEU. The sensor signals are also input to the GPC via FC MDMs for inclusion in the downlist and CRT displays. Limit test for the C&W annunciation are provided by the C&WEU. A description of the logic is as follows:



*The number of inputs to the logic gate vary depending on the system.

B. The Backup (B/U) C&W System

The Backup C&W System (not to be confused with the BFS), is a portion of the BFS software which is used to provide varying degrees of redundancy to the primary C&W system with redundancy being assigned on the basis of function criticality. The B/U system is normally considered to be a software system.



It is comprised of the BFS software that performs limit tests to drive visual and aural alarms when a parameter is out of range from its established limits.

C. The SM alert system

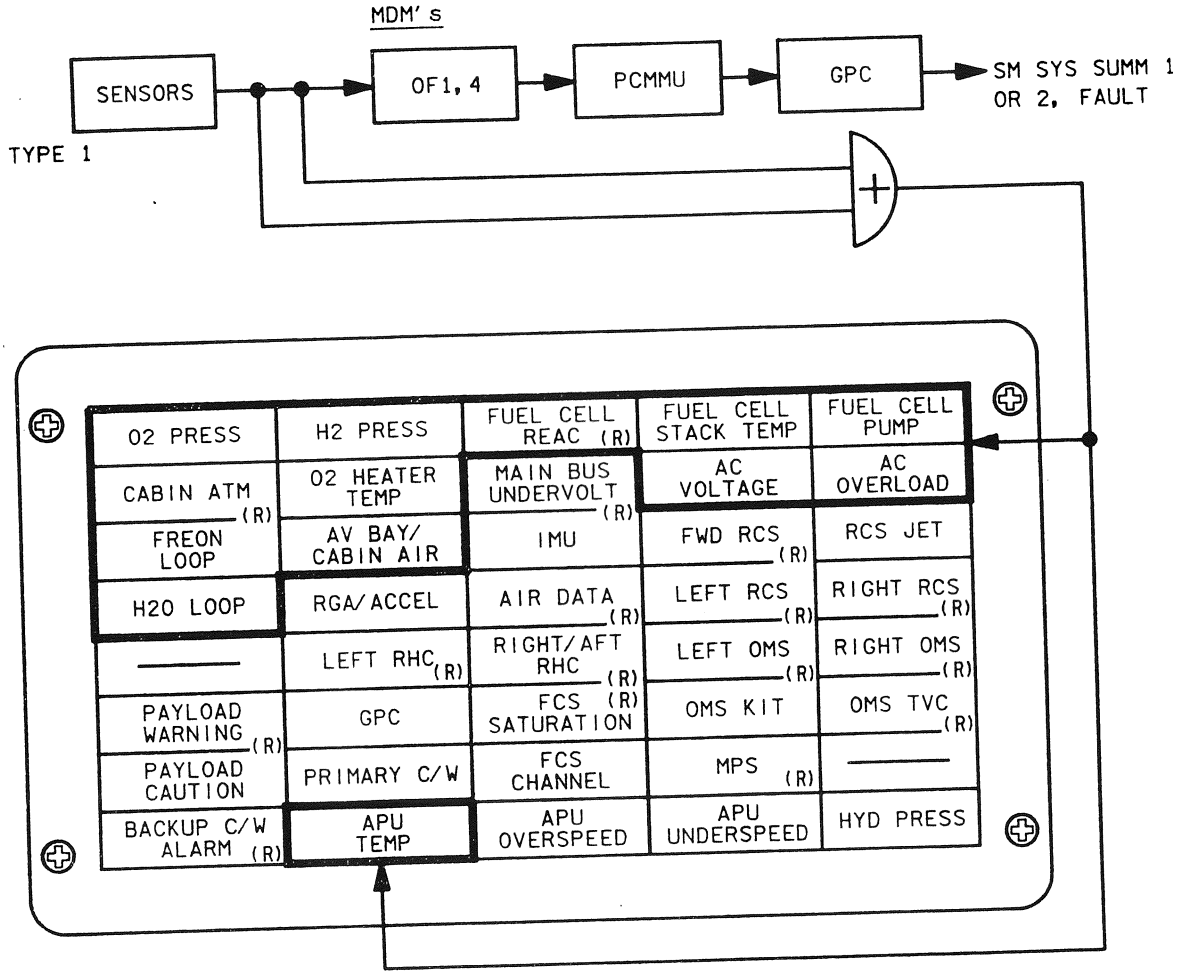
The SM alert function is a portion of the SM program that operates similar to the backup C&W system. The alert program is designed to inform the crew of a situation that may be leading to a C&W situation. When an SM alert parameter exceeds its limits, the SM alert light is illuminated, a discrete is sent to the primary C&W system to turn on the SM tone, and the software displays a CRT fault message on the fault message line and fault summary page.

16.9.1 Types of C&W Functions

There are 11 types of C&W functions in the BFS C&W system of which 6 are applicable to the SM/SP functions. The type of function is based on the approach used to mechanize detection and display of the parameter. The following paragraphs provide a description of each type of function. Following the function type description, there is the type applicable figure together with the lights illuminated in the C&W matrix.

A. Type 1

This function type does not have a GPC Interface and therefore is unaffected by engage or mission modes. Backup C&W information is provided via the same sensors that are used for the hardwired system. This information is provided to the GPC through an OI MDM. The GPC performs limit tests and annunciates the BACKUP C/W ALARM light via interface with the C&WEU when a limit test fails.

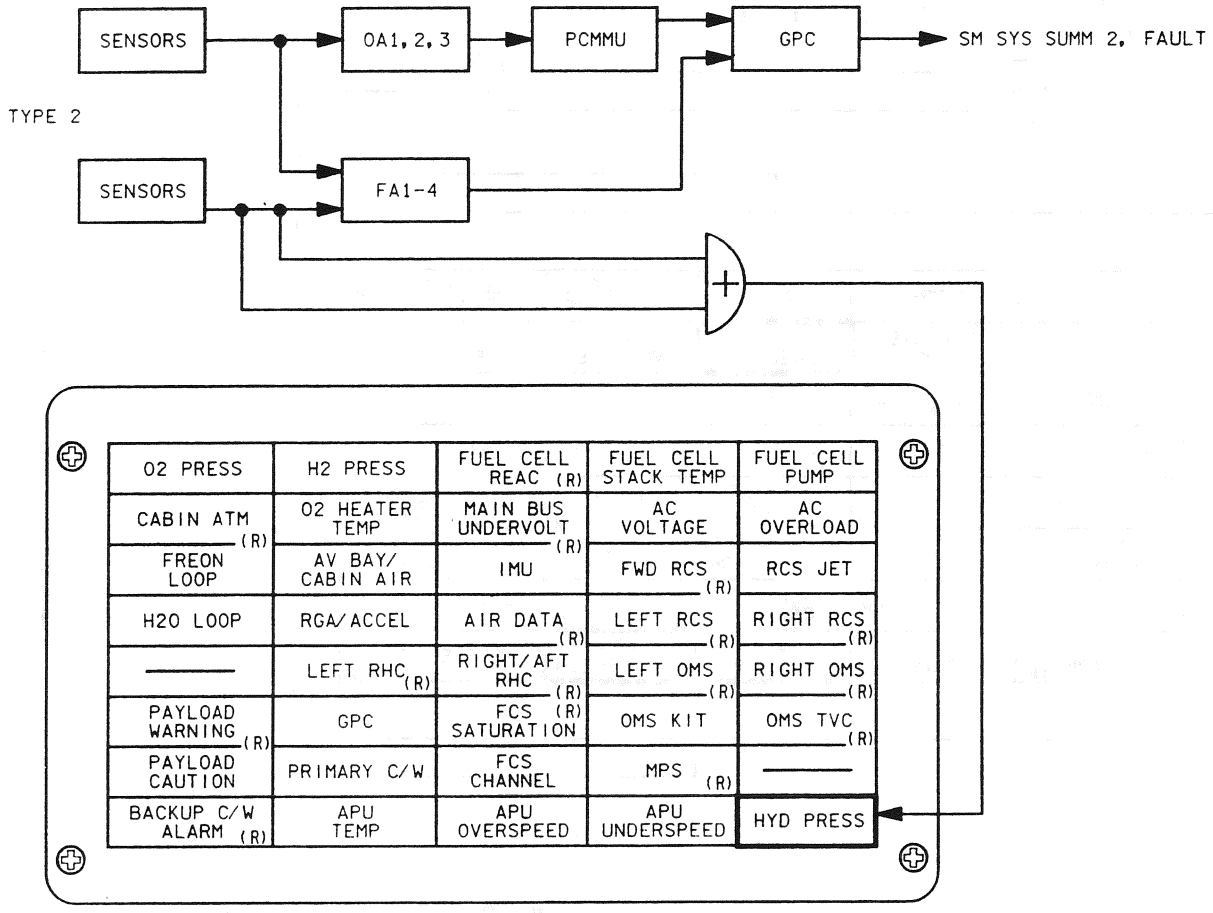


188201239.ART, 1

Figure 16-5.- Type 1 C&W Annunciation.

B. Type 2

This type function is used exclusively for Hydraulic Pressure parameters. This function type does not have a GPC Interface and therefore is unaffected by engage or mission modes. A separate group of sensors is used to provide backup C&W information via OI and FA MDMs to the GPCs. The GPC performs limit test and annunciates the BACKUP C/W ALARM light via interface with the C&WEU through a payload MDM when a limit test fails.



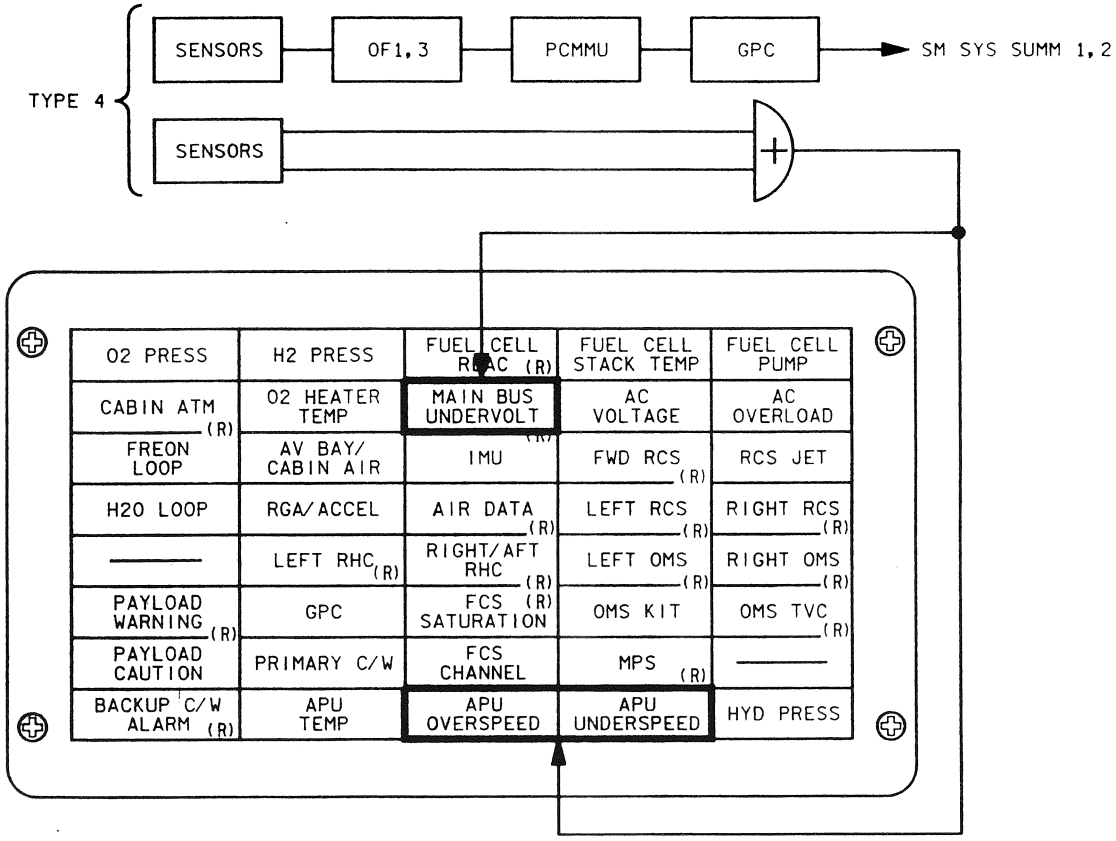
O2 PRESS	H2 PRESS	FUEL CELL REAC (R)	FUEL CELL STACK TEMP	FUEL CELL PUMP
CABIN ATM (R)	O2 HEATER TEMP	MAIN BUS UNDERVOLT (R)	AC VOLTAGE	AC OVERLOAD
FREON LOOP	AV BAY/ CABIN AIR	IMU	FWD RCS (R)	RCS JET
H2O LOOP	RGA/ACCEL	AIR DATA (R)	LEFT RCS (R)	RIGHT RCS (R)
———	LEFT RHC (R)	RIGHT/AFT RHC (R)	LEFT OMS (R)	RIGHT OMS (R)
PAYLOAD WARNING (R)	GPC	FCS (R) SATURATION	OMS KIT	OMS TVC (R)
PAYLOAD CAUTION	PRIMARY C/W	FCS CHANNEL	MPS (R)	———
BACKUP C/W ALARM (R)	APU TEMP	APU OVERSPEED	APU UNDERSPEED	HYD PRESS

188201240.ART, 2

Figure 16-6.- Type 2 C&W Annunciation.

C. Type 4

This function type does not have a GPC Interface and therefore is unaffected by engage or mission modes. Backup C&W information is provided by related (but not the same) performance parameters that are routed to the GPC via OI MDMs. The GPC performs limit tests and annunciates the BACKUP C/W ALARM light via interface with the C&WEU when a limit test fails. These related performance parameters are also provided as supportive data via a CRT display.

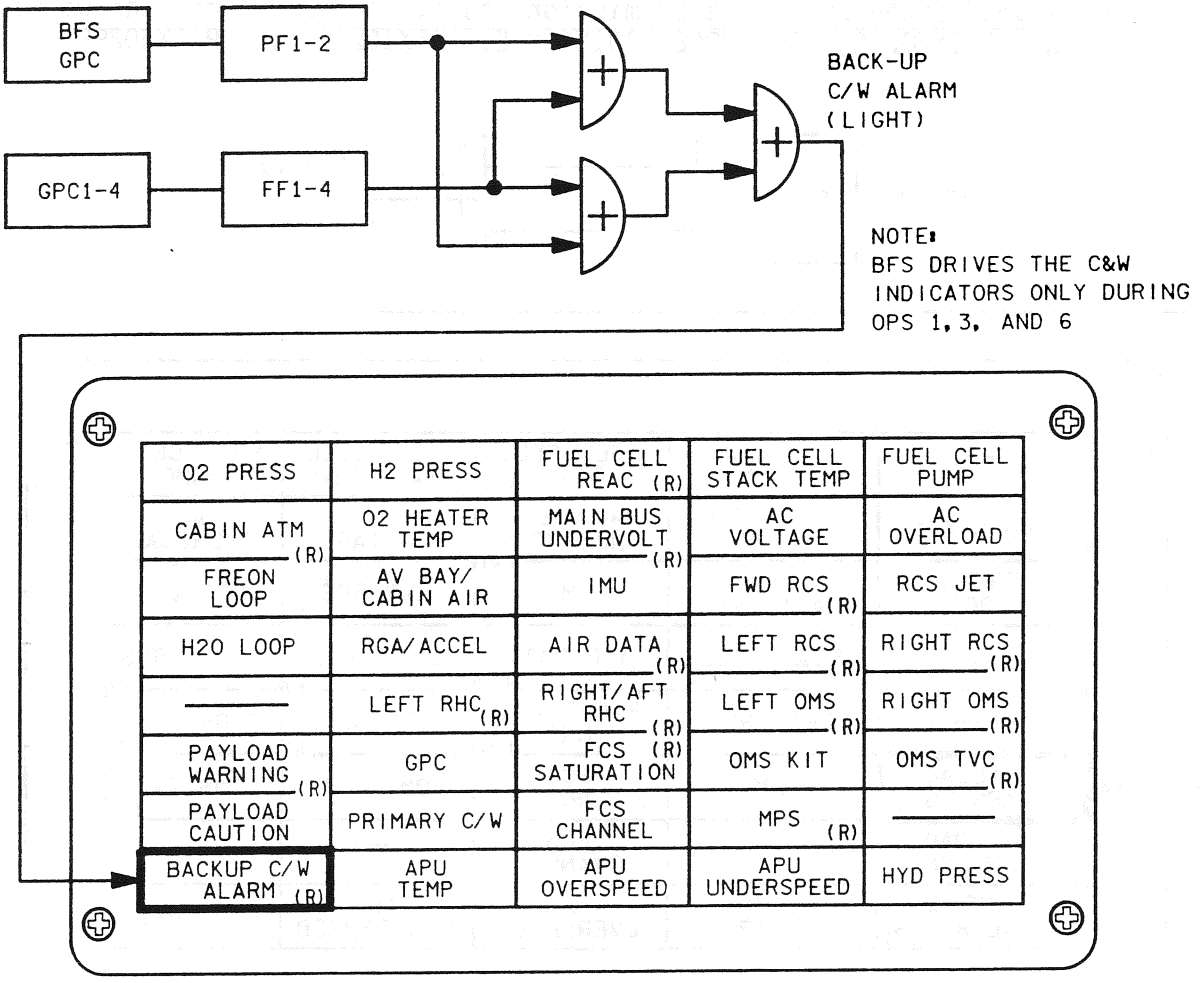


188201235.ART,2

Figure 16-7.- Type 4 C&W Annunciation.

D. Type 9

This type function is used to drive the BACKUP C/W ALARM light via a GPC Interface. In both the preengaged and engaged modes, the BFS software drives this light using the payload buses.

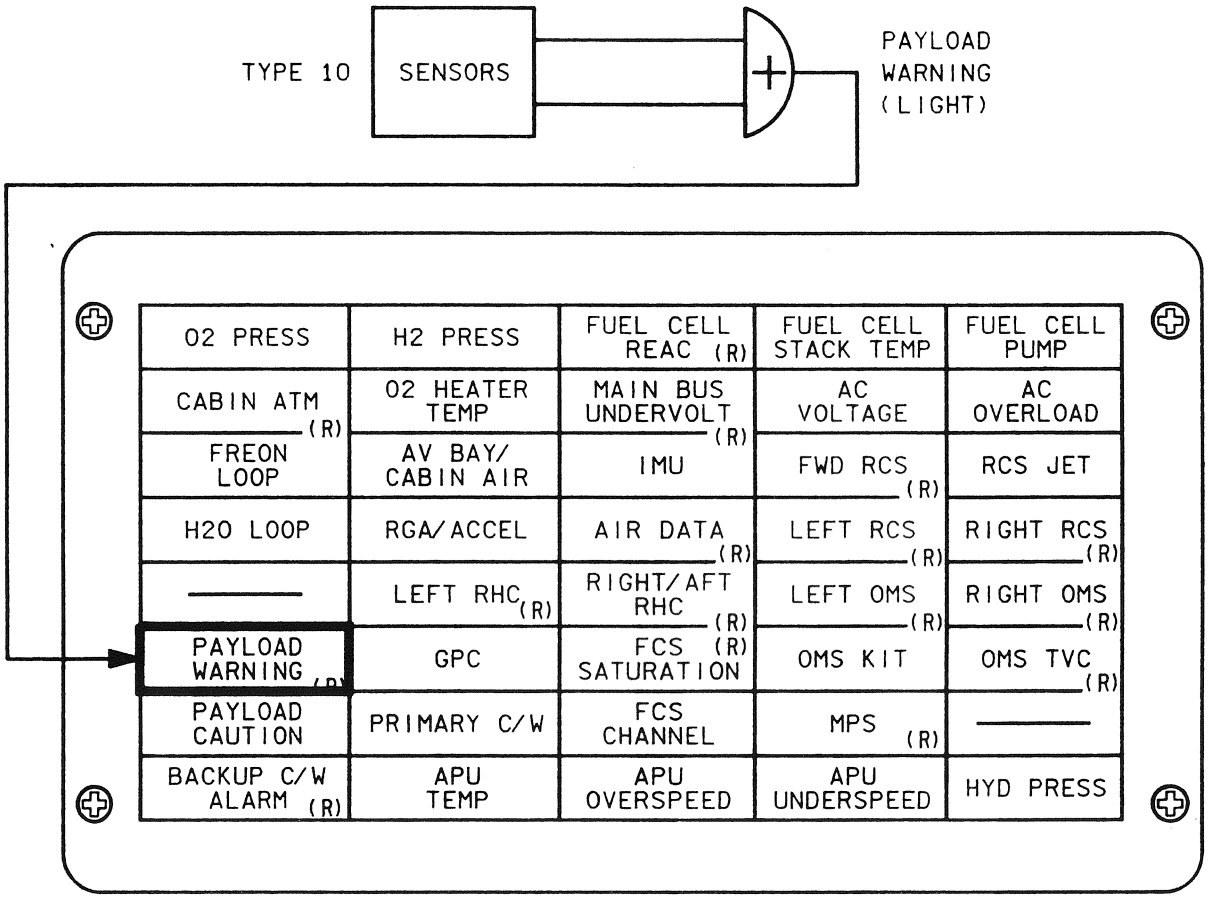


188201236.ART; 2

Figure 16-8.- Type 9 C&W Annunciation.

E. Type 10

This type function is used exclusively to indicate a payload warning. Five hardwired dedicated circuits are used to drive a specific indicator on the C&W annunciator light panel (PAYLOAD WARNING). The circuits are available for all flights but may or may not be connected for any specific flight. This function type does not have a GPC Interface and therefore is unaffected by engage or mission modes. The 5 payload hardwired MSIDs are P01X5025E, P01X5026E, P01X5027E, P01X5028E, and P01X5029E.



188201237.ART, 2

Figure 16-9.- Type 10 C&W Annunciation.

F. Type 11

This type function is used exclusively to indicate a payload caution via a GPC Interface. In both the preengaged and engaged modes the BFS software drives this light. Payload SM performs limit tests or discrete detection. The results are used to drive the PAYLOAD CAUTION light when a limit test fails or a discrete is set, via interface with the C&WEU through a FC MDM. The results are also used to drive the BACKUP C/W ALARM light via the interface with the C&WEU through a payload MDM.

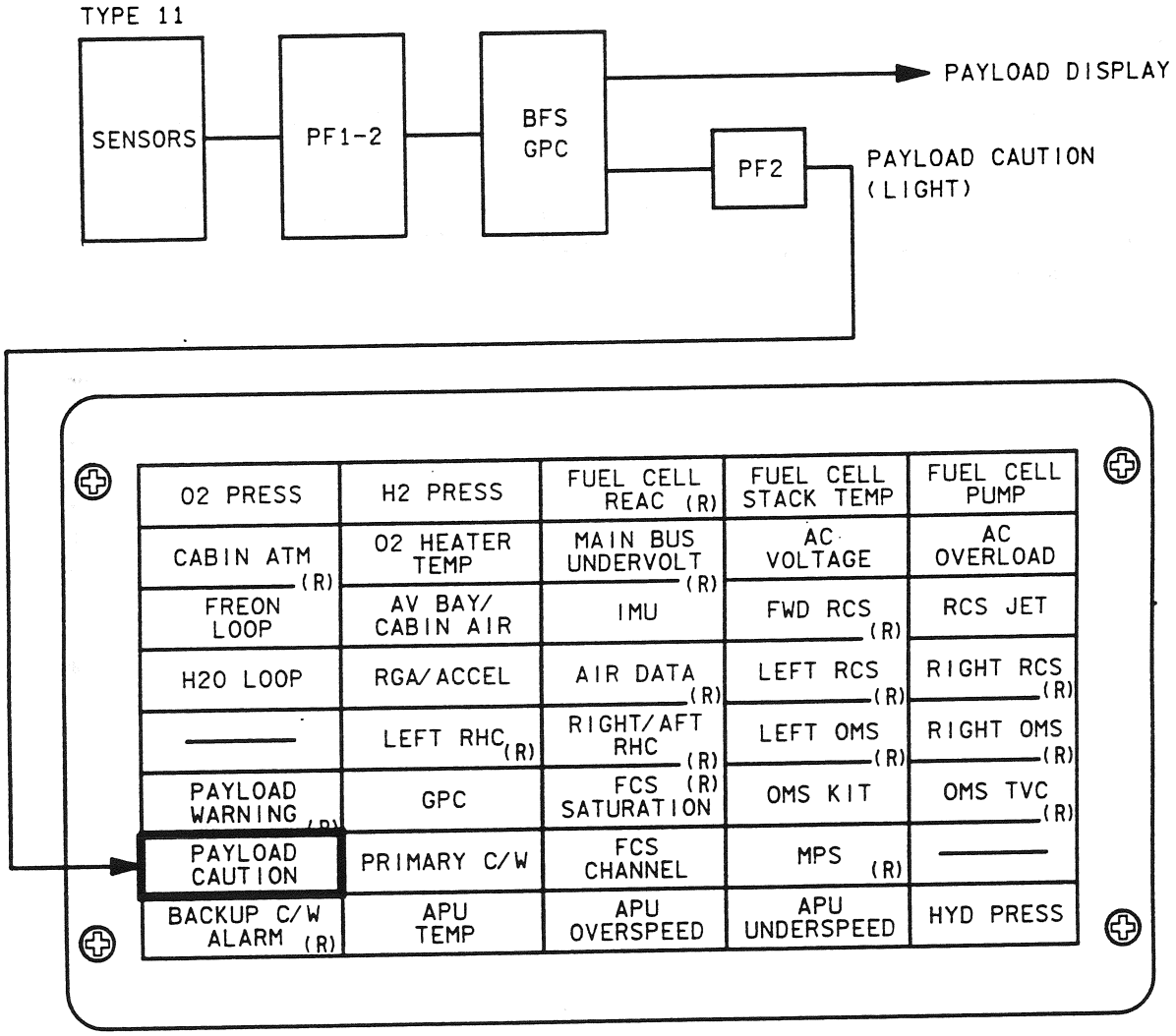


Figure 16-10.- Type 11 C&W Annunciation.

In the pre-engaged mode, the PFS illuminates the SM alert light and initiates the SM tone for certain Class 3 parameters using the flight critical buses, while the BFS illuminates the same SM alert light and initiates the SM tone for certain Class 3 parameters using the payload buses.

Table 16-I contains pertinent information for each flight in the C&W panel applicable to the SM/SP BFS function. However, the system limits specified in the table is representative of what the limit range should be but may not be applicable for each shuttle flight.

16.10 REFERENCES

1. Backup Flight System User's Guide (STS 81-0070A Book 1) dated January 1986

2. Backup Flight System (BFS) Systems Management/Special Processes and Payload Program Requirements Document (MG038119, dated September 8, 1989).

TABLE 16-II.- C&W PARAMETER DATA[1]

C&W annunciator light	Sub-system	MML ID	FUNCTION	System transducer		Hardware C&W limits (EUs)		Hardware C&W limits (volts)		Software C&W limits (EUs)	
				Low	High	Low	High	Low	High	Low	High
O_2 PRESS.	EPG	V46P1110A	O_2 TK 1 PRESS.	515 psia	1015 psia	540 psia	985 psia	0.25	4.70	540 psia	985 psia
O_2 PRESS.		V46P1210A	O_2 TK 2 PRESS.	515 psia	1015 psia	540 psia	985 psia	0.25	4.70	540 psia	985 psia
O_2 PRESS.		V46P1310A	O_2 TK 3 PRESS.	515 psia	1015 psia	540 psia	985 psia	0.25	4.70	540 psia	985 psia
O_2 PRESS.		V46P1410A	O_2 TK 4 PRESS.	515 psia	1015 psia	540 psia	985 psia	0.25	4.70	540 psia	985 psia
O_2 PRESS.		V72	O_2 KIT PRESS.	NOMINAL (OV)	DISCRETE	N/A	5V/28V	5.00	2.50		
H_2 PRESS.	EPG	V46P2110A	H_2 TK 1 PRESS.	145 psia	305 psia	153 psia	293.8 psia	0.25	4.65	153 psia	293.8 psia
H_2 PRESS.		V45P2210A	H_2 TK 2 PRESS.	145 psia	305 psia	153 psia	293.8 psia	0.25	4.65	153 psia	293.8 psia
H_2 PRESS.		V45P2310A	H_2 TK 3 PRESS.	145 psia	305 psia	153 psia	293.8 psia	0.25	4.65	153 psia	293.8 psia
H_2 PRESS.		V45P2410A	H_2 TK 4 PRESS.	145 psia	305 psia	153 psia	293.8 psia	0.25	4.65	153 psia	293.8 psia
H_2 PRESS.		V72	H_2 KIT PRESS.	NOMINAL (OV)	DISCRETE	N/A	5V/28V	5.00	2.50		
FUEL CELL REAC	EPG	V45X1150E	FC1 O_2 REAC VLV OPEN	CLOSED (OV)	OPEN (28V)	OV	N/A	4.80	0.10	CLOSED (0 Vdc)	
FUEL CELL REAC		V45X2150E	FC1 H_2 REAC VLV OPEN	CLOSED (OV)	OPEN (28V)	OV	N/A	4.80	0.10	CLOSED (0 Vdc)	
FUEL CELL REAC		V45X1155E	FC2 O_2 REAC VLV OPEN	CLOSED (OV)	OPEN (28V)	OV	N/A	4.80	0.10	CLOSED (0 Vdc)	
FUEL CELL REAC		V45X2155E	FC2 H_2 REAC VLV OPEN	CLOSED (OV)	OPEN (28V)	OV	N/A	4.80	0.10	CLOSED (0 Vdc)	
FUEL CELL REAC		V45X1160E	FC3 O_2 REAC VLV OPEN	CLOSED (OV)	OPEN (28V)	OV	N/A	4.80	0.10	CLOSED (0 Vdc)	
FUEL CELL REAC		V45X2160E	FC3 H_2 REAC VLV OPEN	CLOSED (OV)	OPEN (28V)	OV	N/A	4.80	0.10	CLOSED (0 Vdc)	
FUEL CELL STACK TEMP	EPG	V46T0120A	FC1 STACK OUT TEMP	-75°F	+300°F	273°F	144°F	3.30	4.25	172.5°F	243.7°F
FUEL CELL STACK TEMP		V45T0220A	FC2 STACK OUT TEMP	-75°F	+300°F	173°F	244°F	3.30	4.25	172.5°F	243.7°F
FUEL CELL STACK TEMP		V45T0320A	FC3 STACK OUT TEMP	-75°F	+300°F	173°F	244°F	3.30	4.25	172.5°F	243.7°F
FUEL CELL PUMP	EPG	V45X0143E	FC1 COOLANT PUMP STAT	OFF (OV)	ON (28V)	OV	N/A	4.80	0.10	OFF (0 Vdc)	
FUEL CELL PUMP		V45X0243E	FC2 COOLANT PUMP STAT	OFF (OV)	ON (28V)	OV	N/A	4.80	0.10	OFF (0 Vdc)	
FUEL CELL PUMP		V45X0343E	FC3 COOLANT PUMP STAT	OFF (OV)	ON (28V)	OV	N/A	4.80	0.10	OFF (0 Vdc)	
CABIN ATM	ECLSS	V61P2406A	CABIN PRESSURE	0 psia	+20 psia	13.8 psia	15.2 psia	3.45	3.80	14.1 psia	15.3 psia
CABIN ATM		V61R2106A	SYS 1 O_2 FLOW RATE	0 pph	5 pph	N/A	4.9 pph	5.00	4.90		5 lb/hr
CABIN ATM		V61R2205A	SYS 2 O_2 FLOW RATE	0 pph	5 pph	N/A	4.9 pph	5.00	4.90		5 lb/h4
CABIN ATM		V61P2511A	O2 PART PRESS. A	0 psia	5 psia	2.8 psia	3.6 psia	2.70	3.45	2.7 psia 2.4 psia*	3.6 psia
CABIN ATM		V61P2513A	O2 PART PRESS. B	0 psia	5 psia	2.8 psia	3.6 psia	2.70	3.45	2.7 psia 2.4 psia*	3.6 psia
CABIN ATM		V61R2553A	SYS 1 GN_2 FLOW RATE	0 pph	5 pph	N/A	4.9 pph	5.00	4.90		
CABIN ATM		V61R2554A	SYS 2 GN_2 FLOW RATE	0 pph	5 pph	N/A	4.9 pph	5.00	4.90		5 lb/hr
O_2 HEATER PUMP	EPG	V45T1107A	O_2 TK 1 HTR 1 TEMP	-425°F	-475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER PUMP		V45T1109A	O_2 TK 1 HTR 2 TEMP	-425°F	-475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER PUMP		V45T1207A	O_2 TK 2 HTR 1 TEMP	-425°F	-475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER PUMP		V45T1209A	O_2 TK 2 HTR 2 TEMP	-425°F	-475°F	N/A	349°F	5.00	4.30		349°F

Note: All C&W channels are dual limit circuits. Limits are established in the C&W electronics unit PROM. Therefore, to use a channel for a high-limit-only channel, both limits are high-limit tested with a nonprogrammable low limit of 5.0 volts. For a low-limit-only channel, both limits are low-limit tested with a nonprogrammable high limit of 0.1 volt. This information is provided as a guide when limit values are measured at panel R13.

*Postflight only

TABLE 16-II.- Concluded

C&W annunciator light	Sub-system	MML ID	FUNCTION	System transducer		Hardware C&W limits (EUs)		Hardware C&W limits (volts)		Software C&W limits (EUs)	
				Low	High	Low	High	Low	High	Low	High
O_2 HEATER TEMP	EPG	V45T1307A	O_2 TK 3 HTR 1 TEMP	-425°F	+475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER TEMP		V45T1309A	O_2 TK 3 HTR 2 TEMP	-425°F	+475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER TEMP		V45T1409A	O_2 TK 4 HTR 1 TEMP	-425°F	+475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER TEMP		V45T1409A	O_2 TK 4 HTR 2 TEMP	-425°F	+475°F	N/A	349°F	5.00	4.30		349°F
O_2 HEATER TEMP		472	O_2 KIT HTR TEMP	NOMINAL (OV)	DISCRETE	N/A	5V/28V	5.00	2.50		
MAIN BUS UNDERVOLT	EPDC	V76V0101A	MN A BUS UNDERVOLT	OV	40V	26.4V	N/A	3.30	0.10	26.4 Vdc	
MAIN BUS UNDERVOLT		V76V0201A	MN B BUS UNDERVOLT	OV	40V	26.4V	N/A	3.30	0.10	26.4 Vdc	
MAIN BUS UNDERVOLT		V76V0301A	MN C BUS UNDERVOLT	OV	40V	26.4V	N/A	3.30	0.10	26.4 Vdc	
AC VOLTAGE	EPDC	V76X1505E	AC1 BUS O/U VOLT	NORMAL (OV)	O/U VOLT (28V)	N/A	28V	5.00	5.00	108 Vac	123 Vac
AC VOLTAGE		V76X1605E	AC2 BUS O/U VOLT	NORMAL (OV)	O/U VOLT (28V)	N/A	28V	5.00	5.00	108 Vac	123 Vac
AC VOLTAGE		V76X1705E	AC3 BUS O/U VOLT	NORMAL (OV)	O/U VOLT (28V)	N/A	28V	5.00	5.00	108 Vac	123 Vac
AC OVERLOAD	EPDC	V76X1506E	AC1 BUS OVERLOAD	NORMAL (OV)	O/LOAD (28V)	N/A	28V	5.00	5.00		28 Vdc
AC OVERLOAD		V76X1606E	AC2 BUS OVERLOAD	NORMAL (OV)	O/LOAD (28V)	N/A	28V	5.00	5.00		28 Vdc
AC OVERLOAD		V76X1706E	AC3 BUS OVERLOAD	NORMAL (OV)	O/LOAD (28V)	N/A	28V	5.00	5.00		28 Vdc
FREON LOOP	ECLSS	V63R1100A	FREON LOOP 1 FLOW RATE	0 pph	2600 pph	1200 pph	N/A	1.00	0.10	1200 pph	
FREON LOOP		V63T1207A	FREON LOOP 1 SINK TEMP	+20°F	+100°F	+32°F	60°F	0.75	2.85	+32°F	100°F
FREON LOOP		V63R1300A	FREON LOOP 2 FLOW RATE	0 pph	2600 pph	1200 pph	N/A	1.00	0.10	1200 pph	
FREON LOOP		V63T1407A	FREON LOOP 2 SINK TEMP	+20°F	+100°F	+32°F	60°F	0.75	2.85	+32°F	100°F
AV BAY/CABIN AIR	ECLSS	V81P2556A	CABIN FAN DELTA PRESS	0 in H_2O	8.0 in H_2O	2.8 in. H_2O	7.04 in. H_2O	1.75	4.40	2.8 in H_2O	7.04 in. H_2O
AV BAY/CABIN AIR		V81P2645A	AV BAY 1 AIR OUT TEMP	+45°F	+145°F	N/A	130°F	5.00	4.25		139°F
AV BAY/CABIN AIR		V81P2650A	AV BAY 2 AIR OUT TEMP	+45°F	+145°F	N/A	130°F	5.00	4.25		139°F
AV BAY/CABIN AIR		V81P2661A	AV BAY 3 AIR OUT TEMP	+45°F	+145°F	N/A	130°F	5.00	4.25		139°F
AV BAY/CABIN AIR		V81P2635A	CABIN HX AIR OUT TEMP	+45°F	+145°F	N/A	130°F	5.00	5.00		65°F
H_2O LOOP	ECLSS	V81P2600A	LOOP 1 H_2O PUMP PRESS	0 psia	150 psia	19.5 psia	79.5 psia	0.65	2.65		79 psia
H_2O LOOP		V81P2700A	LOOP 2 H_2O PUMP PRESS	0 psia	150 psia	45 psia	81.0 psia	1.50	2.70		80/79 psia
PAYLOAD WARNING	P/L	P01X5025E	P/L WARNING	OV	5V/28V	N/A	2.5 V	5.00	2.50		
PAYLOAD WARNING		P01X5026E	P/L WARNING	OV	5V/28V	N/A	2.5 V	5.00	2.50		
PAYLOAD WARNING		P01X5027E	P/L WARNING	OV	5V/28V	N/A	2.5 V	5.00	2.50		
PAYLOAD WARNING		P01X5028E	P/L WARNING	OV	5V/28V	N/A	2.5 V	5.00	2.50		
PAYLOAD WARNING		P01X5029E	P/L WARNING	OV	5V/28V	N/A	2.5 V	5.00	2.50		
PAYLOAD CAUTION	P/L	V72X0001Y	P/L CAUTION	NORMAL (OV)	DISCRETE	N/A	5V	5.00	2.50		

Note: All C&W channels are dual limit circuits. Limits are established in the C&W electronics unit PROM. Therefore, to use a channel for a high-limit-only channel, both limits are high-limit tested with a nonprogrammable low limit of 5.0 volts. For a low-limit-only channel, both limits are low-limit tested with a nonprogrammable high limit of 0.1 volt. This information is provided as a guide when limit values are measured at panel R13.

*Postflight only

APPENDIX A

A

APPENDIX

APPENDIX A
SELF TEST PROGRAM (STP) ERROR MESSAGES

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
R/M REGISTER IS NOT ZERO AFTER A MASTER RESET **ERROR CODE: 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=R/M Status Register R3=0814 0000 R4=0000 0000.	A Master Reset IOP has failed to set the Redundancy Management (R/M) Status Register to zero. The CPU issues the Master Reset prior to starting the self-test, to place the IOP into a known configuration. Part of this configuration calls for the R/M Status register to be zeroed. The self-test program checks all bits in the register for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. R/M Status Register format: (0-no failure, 1-failure) Bit 0=Fail/Timeout Latch Bit 1=Inhibit fail vote inputs during self-test Bit 2=Inhibit fail vote outputs during self-test Bits 3-6=Failure vote inputs 1-4 Bits 7-10=Failure vote outputs 1-4 Bits 11-14=self-test failure votes 1-4 Bit 15=Voter Fail Latch Bit 16=Timeout Latch Bit 17=Voter Termination Control Latch Bit 18=Timer Termination Control Latch Bit 19=IOP Transmission Termination Bits 20-31=Watchdog Timer value.	Failure of the IOP, the R/M logic or the R/M Status Register.
C/M IDLE NOT SET AFTER MASTER RESET **ERROR CODE: 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=INTA R3=0800 0000.	A Master Reset IOP has failed to set the Control/Monitor (C/M) Idle bit to '1' in Interrupt Register A (INTA, bit 2). The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for the C/M Idle bit to be set, which signifies that a C/M Idle Interrupt occurred following the Master Reset, and the C/M logic in the IOP is available for use. The self-test program reads INTA using a Program Controlled Input (PCI) instruction and checks bit 2 for a value of '1'. If it equals '0', the self-test program will annunciate the error and continue.	Failure of the IOP, INTA or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
INT REG A_BITS 1-15 NOT 0 AFTER A MASTER RESET **ERROR CODE: 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=INTA.	A Master Reset IOP has failed to set the significant bits in Interrupt Register A (INTA) to zero. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0,1 and 3-5 of this register to be zeroed, to clear the indication of any External 0 Interrupt. The self-test program reads INTA using a Program Controlled Input (PCI) instruction and masks bits 6-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the IOP, INTA or the Control/Monitor logic.
MASTER RESET DID NOT CLEAR THE TX REGISTER **ERROR CODE: 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=TX register R3=0400 0000.	A Master Reset IOP has failed to disable the MIA Transmitters. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-23 of the MIA Transmitter Enable (TX) Register to be zeroed. This prevents the BCE from transmitting commands or data over the bus to which its MIA is connected. The self-test program reads the TX Register using a Program Controlled Input (PCI) instruction and masks bits 24-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. MIA Transmitter Enable Register format: (0-disabled, 1-enabled) Bits 0-23=MIA Transmitters 1-24, respectively Bits 24-31=Undefined.	Failure of the IOP, the MIA hardware, the TX register or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MASTER RESET DID NOT CLEAR THE RX REGISTER **ERROR CODE: 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=RX register R3=0404 0000.	A Master Reset IOP has failed to disable the MIA Receivers. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-23 of the MIA Receiver Enable (RX) Register to be zeroed. This prevents the BCE from transferring any commands or data it receives from its bus to the BCE. The self-test program reads the RX Register using a Program Controlled Input (PCI) instruction and masks bits 24-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. MIA Receiver Enable Register format: (0-disabled, 1-enabled) Bits 0-23=MIA Receivers 1-24, respectively Bits 24-31=Undefined.	Failure of the IOP, the MIA hardware, the RX register or the Control/Monitor logic.
MASTER RESET DID NOT CLEAR THE DO REGISTER **ERROR CODE: 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=DO register R3=0408 0000.	A Master Reset IOP has failed to disable IPL Discrete Output for the IPL Indicator. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bit 31 of the Discrete Output Status (DO) Register to be zeroed. This signifies that the IPL process is complete or has not been requested. The self-test program reads the DO Register using a Program Controlled Input (PCI) instruction and masks bits 0-30, then checks bit 31 for a value of '0'. If it equals '1', the self-test program will annunciate the error and continue.	Failure of the IOP, the DO register or the Control/ Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MASTER RESET DID NOT CLEAR THE HALT REGISTER **ERROR CODE: 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=Processor Halt register R3=040C 0000.	A Master Reset IOP has failed to set the significant bits in the Status 5 (Processor Halt) Register to zero. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-25 of this register to be zeroed. This disables all processors (MSC and BCE) and terminates any program which the processor was currently executing. The self-test program reads the Processor Halt Register using a Program Controlled Input (PCI) instruction and masks bits 26-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Processor Halt Register format: (0-disabled, 1-enabled) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the IOP, the Status 5 register or the Control/Monitor logic.
MASTER RESET DID NOT CLEAR THE INTB REGISTER **ERROR CODE: 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=INTB R3=0804 0000.	A Master Reset IOP has failed to set the significant bits in Interrupt Register B (INTB) to '0'. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-5 of this register to be zeroed. This clears the indication of any External 1 Interrupt. The self-test program reads INTB using a Program Controlled Input (PCI) instruction and masks bits 3 and 6-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP, INTB or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MASTER RESET DID NOT CLEAR THE INTC REGISTER **ERROR CODE: 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=INTC R3=0808 0000.	A Master Reset IOP has failed to set the significant bits in Interrupt Register C (INTC) to zero. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-11 of this register to be zeroed. This clears the indication of any External 2 Interrupts. The self-test program reads INTC using a Program Controlled Input (PCI) instruction and masks bits 12-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Interrupt Register C format: (0-no interrupt, 1-interrupt) Bits 0-11=MSC Program Interrupts 1-12, respectively Bits 12-31=Zeroes (0).	Failure of the IOP, INTC or the Control/Monitor logic.
MASTER RESET DID NOT CLEAR THE INTD REGISTER **ERROR CODE: 10**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=INTD R3=080C 0000.	A Master Reset IOP has failed to set the significant bits in Interrupt Register D (INTD) to zero. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for this register to be zeroed. This clears the indication of any External 3 Interrupt (Not utilized in the AP-101B). The self-test program reads INTD using a Program Controlled Input (PCI) instruction and masks bits 4-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Interrupt Register D format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the IOP, INTD or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MASTER RESET DID NOT CLEAR THE INTE REGISTER **ERROR CODE: 1 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=INTE R3=0810 0000.	A Master Reset IOP has failed to set the significant bits in Interrupt Register E (INTE) to zero. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for this register to be zeroed. This clears the indication of any External 4 Interrupt (Not utilized in the AP-101B). The self-test program reads INTE using a Program Controlled Input (PCI) instruction and masks bits 4-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Interrupt Register E format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the IOP, INTE or the Control/Monitor logic.
MASTER RESET DID NOT CLEAR THE GO/NO-GO REGISTER **ERROR CODE: 1 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=Go/No-Go register R3=1000 0000. NOTE: R2 value on GPCIPL will be inverted.	A Master Reset IOP has failed to set all of the significant bits in the Status 1 (Go/No-Go) Register to one. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-24 of this register to be set to one. This signifies that the associated processor (MSC or BCE) has not encountered any errors and is in Go mode. The self-test program reads the Go/No-Go Register using a Program Controlled Input (PCI) instruction, masks bits 26-31 and reverses the value of each bit position, then checks all bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Go/No-Go Register format: (0-NoGo, 1-Go) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the IOP, the Go/No-Go register or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MASTER RESET DID NOT CLEAR THE BSY/WAIT REGISTER **ERROR CODE: 1 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=Busy/Wait register R3=1004 0000.	A Master Reset IOP has failed to set the significant bits in the Status 4 (Busy/Wait) Register to zero. The CPU issues the Master Reset prior to starting the self-test program, to place the IOP into a known configuration. Part of this configuration calls for bits 0-24 of this register to be zeroed. This signifies that none of the processors (MSC or BCE) are currently performing a program and are in wait mode. The self-test program reads the Busy/Wait Register using a Program Controlled Input (PCI) instruction and masks bits 26-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Busy/Wait Register format: (0-wait, 1-busy) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the IOP, the Busy/Wait register or the Control/Monitor logic.
AN EXPECTED C/M IDLE INTERRUPT DID NOT OCCUR **ERROR CODE: 1 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3 or R6=C/M Idle interrupt return address.	A Master Reset IOP has failed to cause a C/M (Control/Monitor) Idle Interrupt. The CPU issues the Master Reset prior to starting the self-test program, after a power transient and during the memory unprotect routine, to place the IOP into a known configuration. Part of this configuration calls for this External 0 (EX0) Interrupt to be generated. It signifies that the Control/ Monitor logic is idle and available for use. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of '100C', which allows all pending External 0 interrupts to be annunciated. The value '100C' replaces bits 32-47 of the current PSW, thus setting the EX0 mask (bit 35) to '1'. If the interrupt occurs, the contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX0 PSW (loc. 78-7B). If not, the self-test program will annunciate the error and continue.	Failure of the IOP or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
A PCO_TO RESET BCE HALT BITS_DID NOT RESET ALL **ERROR CODE: 1 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=Processor Halt register R3=040C 0000.	A Program Controlled Output (PCO) instruction from the CPU has failed to place all processors (MSC and BCE) into operate mode. A read of the Status 5 (Processor Halt) Register shows that at least one of the processors remains in a halt mode following execution of the PCO command. The self-test program issues a PCO to the IOP to enable all MSC and BCE Processors, then reads the Processor Halt Status Register, using a Program Controlled Input (PCI) instruction. The Processor Halt Register contains a status of the configuration of all processors. If any of bits 0-25 equal '0', the self-test program will annunciate the error and continue. Processor Halt Register format: (0-disabled, 1-enabled) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the IOP, the Processor Halt register or the Control/Monitor logic.
A PCO_TO ENABLE BCE XMITTERS_DID NOT ENABLE ALL **ERROR CODE: 1 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=TX register R3=0400 0000.	A Program Controlled Output (PCO) instruction from the CPU has failed to enable all of the MIA Transmitters (1-24). A read of the MIA Transmitter Enable (TX) Register shows that one or more of the selected transmitters remains disabled following execution of the PCO command. The self-test program issues a PCO to enable all MIA Transmitters, then reads the TX Register using a Program Controlled Input (PCI) instruction. If any of bits 0-23 equal '0', the self-test program will annunciate the error and continue. TX Register format: (0-disabled, 1-enabled) Bits 0-23=MIA Transmitters 1-24, respectively Bits 24-31=Undefined.	Failure of the IOP, the TX register or the Control/ Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
A PCO_TO ENABLE BCE RECEIVERS_DID NOT ENABLE ALL **ERROR CODE: 1 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=RX register R3=0404 0000.	A Program Controlled Output (PCO) instruction from the CPU has failed to enable all of the MIA Receivers (1-24). A read of the MIA Receiver Enable (RX) Register shows that one or more of the selected receivers remains disabled following execution of the PCO command. The self-test program issues a PCO to enable all MIA Receivers, then reads the RX Register using a Program Controlled Input (PCI) instruction. If any of bits 0-23 equal '0', the self-test program will annunciate the error and continue. RX Register format: (0-disabled, 1-enabled) Bits 0-23=MIA Receivers 1-24, respectively Bits 24-31=Undefined.	Failure of the IOP, the RX register or the Control/ Monitor logic.
MSC OR TESTED BCE IS NO-GO **ERROR CODE: 1 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=Go/No-Go register R3=1000 0000.	The MSC or at least one BCE has encountered an error during execution of the processor self-test routines @STP and #STP. Each routine verifies the proper function of the processors using built-in test instructions. The MSC's addressing logic, data flow, arithmetic logic, parity detection circuitry, channel operation and failure vote logic are tested using the @STP instruction. The BCE's addressing logic, data flow, Direct Memory Access (DMA) logic, channel queues and MIA circuitry are tested using the #STP instruction. Following completion of the CPU test, the program reads the Go/No-Go Register using a Program Controlled Input (PCI) instruction and masks bits 26-31, then checks all other bits for a value of '1'. If any equal '0', the self-test program will annunciate the error and continue. Go/No-Go Register format: (0-NoGo, 1-Go) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the MSC or a BCE.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MSC OR TESTED BCE STILL BUSY **ERROR CODE: 19**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R6=Busy/Wait register R7=1004 0000; 2. R2=Busy/Wait register R3=1004 0000; 3. R7=Busy/Wait register R5=1004 0000.	All processors are not in the Wait State during any of 3 cases: 1. a delay to allow for completion of the MSC/BCE self-test routines @STP and #STP has expired before all processors were complete, 2. the CPU test has completed and at least one IOP processor is still busy executing its self-test, or 3. during the IOP test, a processor has set its Busy/Wait register bit to '1' (Busy) before first being enabled for operation. The program reads the Busy/Wait Register using a Program Controlled Input (PCI) instruction and masks bits 26-31, then checks all other bits for a value of '0'. If any equal '1', the self-test program will annunciate the error and continue. Busy/Wait Register format: (0-wait, 1-busy) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the IOP, the MSC, a BCE or a data flow failure.
AN EXPECTED EXTERNAL INTERRUPT 2 DID NOT OCCUR **ERROR CODE: 20**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The MSC has failed to generate an MSC Program (External 2) interrupt during execution of the processor self-test routines @STP and #STP. Each routine verifies the proper function of the processors using built-in test instructions. The MSC's addressing logic, data flow, arithmetic logic, parity detection circuitry, channel operation and failure vote logic are tested using the @STP instruction. The BCE's addressing logic, data flow, Direct Memory Access (DMA) logic, channel queues and MIA circuitry are tested using the #STP instruction. If the tests complete without error, Interrupt Register C (INTC) bits 0-11 are all set to '1'. If any error is detected during the tests, only select INTC bits are set. Following completion of the CPU test, the program reads the contents of INTC. If it is equal to any value other than 'FFF0 0000', the self-test program will annunciate the error and continue. Interrupt Register C format: (0-no interrupt, 1-interrupt) Bits 0-11=MSC Program Interrupts 1-12, respectively Bits 12-31=Zeroes (0).	Failure of the IOP, the MSC, INTC or the @STP logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MSC COULD NOT TALLY HIGHEST FW MEMORY LOCATION **ERROR CODE: 2 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=MSC Program Counter R3=2201 0000.	The MSC was unable to increment the contents of the highest fullword GPC memory location ('33FFE') and is in a Wait State at an unexpected memory location. This occurs during execution of the MSC/BCE self-test routines @STP and #STP, which verify the proper function of the processors using built-in test instructions. Part of the MSC self-test routine is to execute the Tally and Skip Zero (@TSZ) instruction, which increments the contents of the fullword at location '33FFE' (initialized to 'FFFFFFFF') and skips the next sequential instruction (i.e. a Wait command) if the result of the operation equals zero. Following completion of the CPU test, the program reads the MSC Program Counter, using a Program Controlled Input (PCI) instruction, to verify that the MSC is in a wait state at the correct memory location. If not, the self-test program will annunciate the error and continue.	Failure of the IOP Programmable Read Only Memory (PROM) or the MSC Processor.
AN EXPECTED STORE-PROTECT INTRRPT DIDNT OCCUR **ERROR CODE: 2 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=main memory location tested, R4=contents of the tested memory location.	A Program Check (PC) interrupt was not generated following an attempt to store data into a protected memory location. The CPU should generate a store protect interrupt whenever an attempt is made to write data into a memory location in which the store protect bit is equal to '1'. The self-test program verifies the interrupt logic by reading a memory location, setting the store protect bit to '1' and attempting to write data into the same location. This sequence is repeated for 512 locations throughout GPC main memory. If the PC interrupt occurs on any one of the 512 locations, the CPU will generate an interrupt code of '0007', and store it along with the current Program Status Word (PSW) into low core memory locations 48-4B. The next location will be tested. If the PC interrupt does not occur during execution of any one of the store instructions, the self-test program will annunciate the error, reset the interrupt return address and continue on to test the next location.	Failure of the CPU, the CPU store protect logic or main memory.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
STORE-PROTECT INTRRPT ON AN UNPROTECTED LOCATION **ERROR CODE: 2 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=main memory location tested, R4=contents of the tested memory location.	The CPU erroneously generated a Program Check (PC) interrupt following an attempt to store data into an unprotected memory location. The CPU should not generate a store protect interrupt when an attempt is made to write data into an memory location in which the store protect bit is equal to '0'. The self-test program verifies the interrupt logic by reading a memory location, setting the store protect bit to '0' and storing data into the same location. This sequence is repeated for 512 locations throughout GPC main memory. If the PC interrupt does not occur on any one of the 512 locations, the next location will be tested. If the PC interrupt occurs during execution of any of the store instructions, the CPU will generate an interrupt code of '0007', and store it along with the current Program Status Word (PSW) into low core memory locations 48-4B. The self-test program will then annunciate an error, reset the interrupt return address and continue on to test the next location.	Failure of the CPU, the CPU store protect logic or main memory.
MEMORY ADDRESSING ERROR **ERROR CODE: 2 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=tested address, R3=contents of the reference address inverted R5=reference address R6=contents of the reference address R7=bit position changed in reference address.	The self-test program has determined that a test of the memory addressing logic has failed. The program selects the highest unprotected halfword address of each memory page, referred to as the reference address, and stores it into the Storage Address Register (SAR). Each bit position in the SAR is inverted, one at a time, to produce memory locations whose contents will then be compared to the contents of the reference address. If the values of the 2 locations compare successfully, the value of the reference location is inverted and a second compare is performed. If this compare is also successful, the test is considered failed. The self-test program will annunciate the error and continue onto the next sequential memory page until all have been tested.	Failure of the GPC Memory Addressing logic, main memory or the SAR.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
SYSTEM SOFTWARE LOADER CHECKSUM ERROR **ERROR CODE: 2 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=sum of the contents of the System Software Loader (SSL) routine.	The checksum supplied with the System Software Loader (SSL) does not equal the checksum computed by the CPU. The self-test program, during a default IPL or upon the selection of a PASS software load, will compute a checksum of the SSL prior to transferring control to it. The CPU verifies the data in the SSL by adding each halfword in the routine together and storing it into General Register 4 (R4). The result should equal the value supplied with the SSL, ignoring overflow. If it is not, the self-test program will deschedule the request for the PASS load, annunciate the error and return to the job scheduler.	Incorrect checksum value supplied by the software build, main memory failure or SSL load failure.
MASTER RESET DID NOT CAUSE AN INTERRUPT **ERROR CODE: 2 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=8440 0000.	A Master Reset IOP has failed to cause a C/M (Control/Monitor) Idle Interrupt. The CPU issues the Master Reset prior to transferring control to the System Software Loader (SSL), to place the IOP into a known configuration. Part of this configuration calls for this External 0 (EX0) Interrupt to be generated. It signifies that the Control/Monitor logic is idle and available for use. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'FF0C', which allows all pending External Interrupts to be annunciated. The value 'FF0C' replaces bits 32-47 of the current PSW, thus setting the EX0 mask (bit 35) to '1'. If the interrupt occurs, the contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX0 PSW (loc. 78-7B) and Interrupt Register A (INTA) bit 2 is set to '1'. If not, the self-test program will annunciate the error and continue.	Failure of the IOP, INTA or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MCH CHK: EXTENDED STORAGE ADDRESS PARITY ERROR CODE: 2 7	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the MC PSW.	An unexpected Machine Check (MC) interrupt has been generated while addressing locations in GPC Extended Memory, which refers to memory pages 11-16 (IOP memory) and consists of main memory locations '28000'-'33FFF'. The IOP extended memory unit circuitry has detected bad (even) parity in an address specified by the CPU or IOP for a memory read or write. The IOP, upon detection of the parity error, will send the interrupt signal to the CPU via hardwire. The CPU will then generate a MC interrupt with an interrupt code of '0001', store it along with the current Program Status Word (PSW) into low core locations 40-43 and execute the MC interrupt handler. Since the error can occur during a memory write, the CPU will inhibit the write to prevent an unwanted memory alteration. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU or IOP parity generator circuitry.
MCH CHK: IOP STORAGE PARITY ERROR CODE: 2 8	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the MC PSW.	An unexpected Machine Check (MC) interrupt has been generated while accessing data during an IOP Direct Memory Access (DMA) "out" operation. The IOP parity checker circuitry has detected bad (even) parity in the data read from memory during a DMA 'Out' operation, which is a read of main memory. The IOP, upon detection of the data parity error, will send the interrupt signal to the CPU via hardwire. The CPU will then generate a MC interrupt with an interrupt code of '0002', store it along with the current Program Status Word (PSW) into low core locations 40-43 and execute the MC interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of CPU or IOP main memory.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MCH CHK: CPU STORAGE PARITY **ERROR CODE: 2 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the MC PSW.	An unexpected Machine Check (MC) interrupt has been generated following detection of a parity error during a read of a memory location within the CPU, which refers to memory pages 1-10 and consists of main memory locations '00000'-'27FFF'. The CPU parity checker circuitry has detected bad (even) parity stored in the contents of the memory location. The CPU, upon detection of the storage parity error, will generate a MC interrupt with an interrupt code of '0003', store it along with the current Program Status Word (PSW) into low core locations 40-43 and execute the MC interrupt handler. Since the error can occur during a read-compute-write operation, the CPU will inhibit the write to prevent an unwanted memory alteration. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of CPU main memory.
MCH CHK: EXTENDED STORAGE DATA PARITY **ERROR CODE: 3 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the MC PSW.	An unexpected Machine Check (MC) interrupt has been generated while the CPU was accessing data stored in a location within GPC Extended Memory, which refers to memory pages 11-16 (IOP memory) and consists of main memory locations '28000'-'33FFF'. The IOP parity checker circuitry has detected bad (even) parity in the data read from IOP memory. The IOP, upon detection of the data parity error, will send the interrupt signal to the CPU via hardwire. The CPU will then generate a MC interrupt with an interrupt code of '0004', store it along with the current Program Status Word (PSW) into low core locations 40-43 and execute the MC interrupt handler. Since the error can occur during a read-compute-write operation, the CPU will inhibit the write to prevent an unwanted memory alteration. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of IOP main memory.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MCH CHK: CPU ROS PARITY **ERROR CODE: 3 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the MC PSW.	An unexpected Machine Check (MC) interrupt has been generated while the CPU was attempting to execute a microinstruction read from the CPU Read Only Storage (ROS). The microinstruction, read from ROS and placed in the ROS data register (ROSDR), contained bad (even) parity when output from the ROSDR. Upon detection of the parity error, the CPU will generate a MC interrupt with an interrupt code of '0005', store it along with the current Program Status Word (PSW) into low core locations 40-43 and execute the MC interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU ROS parity generator or the ROS data register.
MCH CHK: INTERRUPT CODE NOT IN THE RANGE 1 - 5 **ERROR CODE: 3 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the MC PSW.	An unexpected Machine Check (MC) interrupt has been generated with an incorrect interrupt code contained in the MC Program Status Word (PSW). The interrupt code is generated by the CPU following detection of an error condition, and is contained in the MC PSW (loc. 40-43). The self-test program verifies the interrupt code by loading the value stored at low core location 43 into General Purpose Register 4 (G4). A check is then made to determine if the returned value is within the valid range of '0001' through '0005'. If not, the self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU interrupt microcode.
PGM CHK: PRIVILEGED INSTRUCTION **ERROR CODE: 3 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) interrupt has been generated while the CPU was attempting to execute a privileged instruction while in the Problem State. The CPU must be in the Supervisor State to execute any privileged instructions. This state is entered by executing a Supervisor Call (SVC) instruction and is signified by bit 47 of the Program Status Word (PSW) being equal to '0'. Privileged instructions include: Insert Store Protect Bit (ISPB); Load Program Status (LPS); Set System Mask (SSM); Internal Control (IC); Program Controlled Input/Output (PCI/O). Upon detection of the error, the CPU will generate a PC interrupt with an interrupt code of '0001', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure to place the CPU in Supervisor State prior to execution of a privileged instruction.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
PGM CHK: INTERRUPT CODE 2 IS NOT DEFINED **ERROR CODE: 3 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) interrupt has been generated with an undefined interrupt code of '0002' contained in the PC Program Status Word (PSW). The interrupt code is generated by the CPU following detection of an error condition, and is contained in the PC PSW at low core locations 48-4B. The PC interrupt handler loads the value stored at low core location 4B into General Purpose Register 4 (G4). A branch instruction is then performed to annunciate the correct error message. The valid range of PC interrupt codes is '0000' through '000C', with '0002' and '0008' undefined. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU interrupt microcode.
PGM CHK: CPU ADDRESS SPECIFICATION ERROR. **ERROR CODE: 3 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) interrupt has been generated while the CPU was performing a main memory access. The CPU has determined that an invalid memory location was specified during a memory read or write. The CPU, upon detection of the address specification error, will generate a PC interrupt with an interrupt code of '0003', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU or incorrect address inputs to a main memory operation.
PGM CHK: FIXED POINT OVERFLOW **ERROR CODE: 3 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) interrupt has been generated following execution of a fixed point operation which resulted in a value too large to be stored in a fullword (32 bit) format. The CPU contains two sets of eight 32-bit registers which are used to perform all fixed point arithmetic operations. Upon detection of the overflow condition, the CPU will generate a PC interrupt with an interrupt code of '0004', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, a fixed point register or incorrect inputs to a fixed point arithmetic operation.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
PGM CHK: SIGNIFICANCE ERROR **ERROR CODE: 3 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated following execution of a floating point operation which resulted in a fractional value equal to zero and a non-zero characteristic. A floating point value contains a sign bit, a characteristic and a fraction. The characteristic is the representation of the exponent in Excess-64 notation and must be zero when the fraction is zero. The CPU contains one set of eight 32-bit registers which are used to perform all floating point arithmetic operations. Upon detection of the significance error, the CPU will generate a PC interrupt with an interrupt code of '0005', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, a floating point register or incorrect inputs to a floating point operation.
PGM CHK: UN-NORMALIZED INPUTS -FLTING POINT DVD- **ERROR CODE: 3 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated prior to execution of a floating point divide instruction in which the inputs to the instruction were unnormalized. A floating point value, containing a sign bit, a characteristic and a fraction, is said to be normalized if the high-order hex digit of the fraction is non-zero. This feature provides the greatest precision possible for floating point operations. Upon detection of the normalization error, the CPU will generate a PC interrupt with an interrupt code of '0006', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, a floating point register or incorrect inputs to a floating point divide operation.
PGM CHK: CPU STORE PROTECT VIOLATION **ERROR CODE: 4 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated while the CPU was performing a main memory write into a protected memory location. The CPU has specified a memory location in which the store protect bit is equal to '1'. Upon detection of the store protect violation, the CPU will inhibit the memory write, generate a PC interrupt with an interrupt code of '0007', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU or incorrect address inputs to a main store operation.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
PGM CHK: INTERRUPT CODE 8 IS NOT DEFINED **ERROR CODE: 4 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	The CPU has generated an unexpected Program Check (PC) interrupt with an undefined interrupt code of '0008' contained in the PC Program Status Word (PSW). The interrupt code is generated by the CPU following detection of an error condition, and is contained in the PC PSW at low core locations 48-4B. The PC interrupt handler loads the value stored at low core location 4B into General Purpose Register 4 (G4). A branch instruction is then performed to annunciate the correct error message. The valid range of PC interrupt codes is '0000' through '000C', with '0002' and '0008' undefined. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU interrupt microcode.
PGM CHK: EXPONENT UNDERFLOW -FLOATING POINT- **ERROR CODE: 4 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated following execution of a floating point operation which resulted in a characteristic that could not be expressed in a 7-bit format. A floating point value contains a sign bit, a characteristic and a fraction. The characteristic is the representation of the exponent in Excess-64 notation and must be in the range of 0-127. Upon detection of the error, the CPU will generate a PC interrupt with an interrupt code of '0009', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, floating point register or incorrect inputs to a floating point operation.
PGM CHK: EXPONENT OVERFLOW -CONVERT INSTRUCTION- **ERROR CODE: 4 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated following execution of a Convert to Floating Point (CVFL) instruction which resulted in a non-zero fraction and a characteristic greater than 127. A floating point value contains a sign bit, a characteristic and a fraction. The characteristic is the representation of the exponent in Excess-64 notation, and must be in the range of 0-127. Upon detection of the error, the CPU will generate a PC interrupt with an interrupt code of '000A', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, floating point register or incorrect inputs to a CVFL instruction.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
PGM CHK: EXPONENT OVERFLOW -FLOATING POINT- **ERROR CODE: 4 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated following execution of a floating point operation which resulted in a non-zero fraction and a characteristic greater than 127. A floating point value contains a sign bit, a characteristic and a fraction. The characteristic is the representation of the exponent in Excess-64 notation, and must be in the range of 0-127. Upon detection of the error, the CPU will generate a PC interrupt with an interrupt code of '000B', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, floating point register or incorrect inputs to a floating point operation.
PGM CHK: DIVIDE INSTRUCTION -FLOATING- **ERROR CODE: 4 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	An unexpected Program Check (PC) Interrupt has been generated prior to execution of a floating point divide instruction in which the input denominator was equal to '0'. Upon detection of the error, the CPU will generate a PC interrupt with an interrupt code of '000C', store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure to provide correct inputs to a floating point divide instruction.
PGM CHK: INTERRUPT CODE GREATER THAN 12 **ERROR CODE: 4 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the PC PSW.	The CPU has generated an unexpected Program Check (PC) interrupt with an undefined interrupt code greater than '000C' contained in the PC Program Status Word (PSW). The interrupt code is generated by the CPU following detection of an error condition, and is contained in the PC PSW at low core locations 48-4B. The PC interrupt handler loads the value stored at low core location 4B into General Purpose Register 4 (G4). A branch instruction is then performed to annunciate the correct error message. The valid range of PC interrupt codes is '0000' through '000C', with '0002' and '0008' undefined. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU interrupt microcode.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED SVC INTERRUPT **ERROR CODE: 47**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=SVC interrupt code.	A Supervisor Call (SVC) Interrupt has been generated with an illegal interrupt code in the Supervisor Call Program Status Word (PSW). This error is due to either of 2 cases: 1) Interrupt code > the number of allowable SVC's in the Self-test program -or- 2) Interrupt code = hex '8000', illegal high-priority SVC 0 interrupt. The SVC interrupt logic is utilized by the self-test program to annunciate any error detected in the CPU, the IOP or main memory. Upon detection of the error, the CPU will generate an SVC interrupt with an interrupt code equal to the error code, store it along with the current Program Status Word (PSW) into low core locations 58-5B and execute the SVC interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU.
UNEXPECTED CLOCK 1 INTERRUPT **ERROR CODE: 48**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the Clock 1 PSW.	An unexpected Program Counter (Clock) 1 Interrupt has been generated while the GPC self-test program was running. The CPU maintains 2 clocks, each composed of one 16-bit binary hardware register and one 16-bit main memory location. The hardware register, with a clock time of 1 microsecond, will cause the associated memory location to be decremented by 1 each time the register passes from '0000' to 'FFFF' (approx. every 65.536 microseconds). When the main memory location counts down to zero, a Clock 1 Interrupt will be generated, the CPU will store the contents of the current Program Status Word (PSW) into low core locations 60-63 and the Clock 1 Interrupt handler will be executed. Clock 1 is used to maintain the 40 millisecond cycle of the computer. The self-test program will annunciate the error, load the current PSW with the return address stored at the time of the interrupt and continue from that point.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED CLOCK 2 INTERRUPT **ERROR CODE: 4 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the Clock 2 PSW.	An unexpected Program Counter (Clock) 2 Interrupt has been generated while the GPC self-test program was running. The CPU maintains 2 clocks, each composed of one 16-bit binary hardware register and one 16-bit main memory location. The hardware register, with a clock time of 1 microsecond, will cause the associated memory location to be decremented by 1 each time the register passes from '0000' to 'FFFF' (approx. every 65.536 microseconds). When the main memory location counts down to zero, a Clock 2 Interrupt will be generated, the CPU will store the contents of the current Program Status Word (PSW) into low core locations 68-6B and the Clock 2 Interrupt handler will be executed. Clock 2 is used to verify the presence of computer control. Each time the clocks are initialized, clock 2 is set to a value .5 microseconds greater than clock 1. A loss of computer control is signified by clock 2 expiring before clock 1. The self-test program will annunciate the error, load the current PSW with the return address stored at the time of the interrupt and continue from that point.	Failure of the CPU.
UNEXPECTED INSTRUCTION MONITOR INTERRUPT **ERROR CODE: 5 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the IM PSW.	An unexpected Instruction Monitor (IM) Interrupt has been generated due to an attempt by the CPU to fetch the contents of an unprotected memory location for use as program code. All program code or instructions must be stored as protected data in memory, and the instruction monitor feature prevents a program from erroneously executing data as instructions. The CPU, upon detection of the instruction monitor violation, will generate an IM interrupt with an interrupt code of '000F', store it along with the current Program Status Word (PSW) into low core locations 70-73 and execute the IM Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU or incorrect store protect bit setting within GPC memory.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXTERNAL 0 INTERRUPT **ERROR CODE: 5 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G3=INTA G4=memory location of the EX0 PSW G7=0800 0000.	An unexpected External 0 (EX0) interrupt has been generated with either invalid bit(s) set in Interrupt Register A (INTA) or no bits set at all. The value of INTA is set by the IOP following detection of an EX0 error condition, and is read by the CPU using a Program Controlled Input (PCI) instruction. The EX0 interrupt handler stores the register value into General Purpose Register 4 (G4) and checks to determine if any of bits 0-4 are set to '1'. If not, it will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the IOP, the Control/Monitor logic or INTA.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXTRNL 1 INTRPT: INTRPT CODE=0 **ERROR CODE: 5 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated with an interrupt code of '0000' stored in the EX1 Program Status Word (PSW), indicating that the IOP has detected an error. An interrupt code of any value other than '0000' indicates that the CPU detected the error. Upon detection of the error, the IOP will set the appropriate bit(s) in Interrupt Register B (INTB) to '1' and send the EX1 interrupt signal to the CPU via hardwire. The CPU will then generate the EX1 interrupt, store the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. INTB must be interrogated to determine the type of error detected. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP.
UNEXPECTED EXTRNL 1 INTRPT: PCI DATA PARITY **ERROR CODE: 5 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated while the IOP was responding to a Program Controlled Input (PCI) request from the CPU. The CPU parity checker circuitry has detected bad (even) parity in the data transferred from the IOP in response to a PCI instruction, which is used by the CPU to read IOP internal registers. The CPU, upon detection of the PCI data parity error, will generate an EX1 interrupt with an interrupt code of '0001', store it along with the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or the parity generator in the Channel Control logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXTRNL 1 INTRPT: I/O WRITE PARITY **ERROR CODE: 5 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated while the IOP was performing a Direct Memory Access (DMA) 'In' operation. A DMA 'In' is a main memory write. The parity checker circuitry has detected bad (even) parity in the data which was to be stored in memory. The CPU, upon detection of the I/O data parity error, will inhibit the memory write, generate an EX1 interrupt with an interrupt code of '0002', store it along with the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or the parity generator in the Channel Control logic.
UNEXPECTED EXTRNL 1 INTRPT: I/O ADDRESS SPEC ERROR **ERROR CODE: 5 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated while the IOP was performing a Direct Memory Access (DMA) operation. DMA's are used by the IOP for main memory reads and writes. The CPU has determined that an invalid memory location was specified by the IOP during the DMA. The CPU, upon detection of the I/O address specification error, will generate an EX1 interrupt with an interrupt code of '0003', store it along with the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP, the DMA Address Register or incorrect address inputs to a DMA operation.
UNEXPECTED EXTRNL 1 INTRPT: I/O STORE-PROTECT **ERROR CODE: 5 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated while the IOP was performing a Direct Memory Access (DMA) 'In' operation into a protected memory location. A DMA 'In' is a main memory write. The IOP has specified a memory location in which the store protect bit is equal to '1'. The CPU, upon detection of the store protect violation, will inhibit the memory write, generate an EX1 interrupt with an interrupt code of '0004', store it along with the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or incorrect inputs to a DMA 'In' operation.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXTRNL 1 INTRPT: I/O ADDRESS PARITY **ERROR CODE: 5 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated while the IOP was performing a Direct Memory Access (DMA) operation. DMA's are used by the IOP for main memory reads and writes. The parity checker circuitry has detected bad (even) parity in the address which was specified for a DMA 'In' (write) or 'Out' (read) operation. The CPU, upon detection of the I/O address parity error, will generate an EX1 interrupt with an interrupt code of '0005', store it along with the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or the parity generator in the Channel Control logic.
UNEXPECTED EXTRNL 1 INTRPT: AGE INTERRUPT **ERROR CODE: 5 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	An unexpected External 1 (EX1) Interrupt has been generated with an interrupt code of '0006' stored in the EX1 Program Status Word (PSW), indicating that the AGE (Air/Ground Equipment) to IOP interface line has been activated. The AGE interface is used to connect the IOP and CPU to tester hardware external to the GPC, for troubleshooting purposes. When activated, tester hardware assumes full control of the computer. Upon detection of the line is active, the IOP will send the EX1 interrupt signal to the CPU via hardwire. The CPU will then generate the EX1 interrupt, store the current Program Status Word (PSW) into low core locations 80-83 and execute the EX1 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or the AGE logic within the IOP.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXTERNAL 2 INTERRUPT **ERROR CODE: 5 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G2=INTC G3=0808 0000 G4=memory location of the EX2 PSW.	An unexpected External 2 (EX2) Interrupt has been generated while the self-test program was running, indicating that an MSC program has unexpectedly executed a programmable Interrupt (@INT) instruction. The EX2 class of interrupts is designed for any MSC program defined function. Prior to execution of the @INT instruction, the MSC program will set the appropriate bit(s) in Interrupt Register C (INTC). The IOP will then send the EX2 interrupt signal to the CPU via hardwire. The CPU will generate the EX2 interrupt, store the current Program Status Word (PSW) into low core locations 88-8B and execute the EX2 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register C format: (0-no interrupt, 1-interrupt) Bits 0-11=MSC Program Interrupts 1-12, respectively Bits 12-31=Zeroes (0).	Failure of the IOP or an MSC program.
UNEXPECTED EXTERNAL 3 INTERRUPT **ERROR CODE: 6 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX3 PSW G7=080C 0000.	An unexpected External 3 Interrupt has been generated during execution of the GPC self-test program (STP). This priority of interrupt is not utilized in the AP-101B and should not occur during GPC IPL or the STP. Upon detection of the error, the IOP will set the appropriate bit(s) in Interrupt Register D (INTD) to '1' and send the EX3 interrupt signal to the CPU via hardwire. The CPU will then generate the EX3 Interrupt, store the current Program Status Word (PSW) into low core locations 90-93 and execute the EX3 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register D format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the CPU, the IOP or INTD.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXTERNAL 4 INTERRUPT **ERROR CODE: 6 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX4 PSW G7=0810 0000.	An unexpected External 4 (EX4) Interrupt has been generated during execution of the GPC self-test program (STP). This priority of interrupt is not utilized in the AP-101B and should not occur during GPC IPL or the STP. Upon detection of the error, the IOP will set the appropriate bit(s) in Interrupt Register E (INTE) to '1' and send the EX4 interrupt signal to the CPU via hardwire. The CPU will then generate the EX4 interrupt, store the current Program Status Word (PSW) into low core locations 98-9B and execute the EX4 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register E format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the CPU, the IOP or INTE.
BCE TIME-OUT N RETRY FAIL-DEUIPL **ERROR CODE: 6 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G2=0808 0000 G3=INTC.	An unexpected External 2 (EX2) Interrupt has been generated and the contents of Interrupt Register C (INTC) indicates that a BCE has timed out on two consecutive attempts to perform a Display Electronics Unit (DEU) load. The MSC will attempt the DEU load twice before terminating the job. Upon detection of the timeout, the MSC program will check to see if the number of retries equals '2'. If not, the load is re-attempted. If the number of retries equals '2', INTC is set to '2000 0000' to indicate a BCE timeout, the DEU load is descheduled and the Interrupt (@INT) instruction is executed to interrupt the CPU. The CPU will then generate the EX2 interrupt, store the current Program Status Word (PSW) into low core locations 88-8B and execute the EX2 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of a BCE or a DEU.
GPC ID VALUE NOT IN RANGE 1-5 **ERROR CODE: 6 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=GPC ID.	The GPC identifier number contained in the Discrete Input B (DIB) register is invalid. The ID is determined by a hardwire connector and is set according to the physical position of the GPC. Each time the DEU poll routine is executed, the CPU issues a Program Controlled Input (PCI) instruction to the IOP to read the DIB register. Bits 0-2 of the register are then checked to verify that they are within the valid range of 1 through 5. If not, the self-test program will annunciate the error and continue.	Failure of the DIB register, the Control/Monitor logic or the GPC hardwire connector.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
CONDITION CODE NOT ZERO AFTER A FULL DETECT TEST **ERROR CODE: 6 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The Condition Code (CC) in the current Program Status Word (PSW) has been set to a value other than '00' following the CPU Full Detect Test. The CC is a 2-bit value within the PSW which is set by the results of certain operations. Once set, the CC is used to control the logical flow of a program. The self-test program utilizes the DETECT (0,0) instruction, which first tests the Power Down Putaway logic, the Local Store register and addressing logic, the arithmetic logic and working registers and the CPU Status register. Any error detected in these tests results in a CC of '11'. It then tests the macrointerrupt, microinterrupt and Read Only Storage (ROS) parity interrupt decode and priority logic. Any error detected in these tests results in a CC of '01', indicating a BITE circuitry failure. The self-test program will annunciate the error and continue.	Failure of the CPU or the BITE circuitry..
CONDITION CODE NOT ZERO AFTER A COUNTERS TEST **ERROR CODE: 6 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The Condition Code (CC) in the current Program Status Word (PSW) has been set to a value other than '00' following the CPU Counters Test. The CC is a 2-bit value within the PSW which is set by the results of certain operations. Once set, the CC is used to control the logical flow of a program. Each counter is made up of one 16-bit decrementing hardware register and one 16-bit memory location. Approximately every 65 microseconds, the hardware register cycles from '0000' to 'FFFF', causing the memory location to be decremented by 1. When the memory location cycles from '0000' to 'FFFF', a counter interrupt is generated. The self-test program utilizes the DETECT (1,0) instruction to verify that the counter can be set to all zeroes and all ones, that an interrupt is generated when the counter cycles from '0000' to 'FFFF and that the counter does decrement. If an error is detected during the test, the CC is set to '11'. The self-test program will annunciate the error and continue.	Failure of the CPU or the counter logic.
REG 4 NOT LOADED CORRECTLY VIA R2-TO-R3- TO-R4 **ERROR CODE: 6 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2 through R6=1111 0000.	The CPU has determined that either General Purpose Register 2,3 or 4 (R2, R3 or R4) has been loaded incorrectly during a test of the Load Register (LR) instruction. The test, made within the arithmetic logic portion of the CPU test, loads a value of '1111 0000' into R2. R3 is then loaded from R2, R4 is loaded from R3 and so on. A compare is then made between R3 and R4 to determine if the registers are equal. If not, the self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
THE SUM FORMED IN R2 IS TOO HIGH **ERROR CODE: 6 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=CCCD 0000 R3=4444 0000 R4=3333 0000 R5=2222 0000 R6=1111 0000.	The CPU has determined that one of the General Purpose Registers 2-6 (R2-R6) contains an incorrect value during a test of several register to register instructions. The test, made within the arithmetic logic portion of the CPU test, utilizes R2-R6 to perform a series of Add, Subtract, Or and Exclusive Or (AR, SR, OR and XR) instructions. The result, stored in R2, is compared to the value 'CCCD 0000'. If the contents of R2 is greater than this value, the self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
THE SUM FORMED IN R2 IS TOO LOW **ERROR CODE: 6 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=CCCD 0000 R3=4444 0000 R4=3333 0000 R5=2222 0000 R6=1111 0000.	The CPU has determined that one of the General Purpose Registers 2-6 (R2-R6) contains an incorrect value during a test of several register to register instructions. The test, made within the arithmetic logic portion of the CPU test, utilizes R2-R6 to perform a series of Add, Subtract, Or and Exclusive Or (AR, SR, OR and XR) instructions. The result, stored in R2, is compared to the value 'CCCD 0000'. If the contents of R2 is less than this value, the self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
THE SHIFT TEST FAILED **ERROR CODE: 6 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=2222 0000 R3=0004 4440 R5=2222 0000.	The CPU has determined that either General Purpose Register 2 or 5 (R2 or R5) contains an incorrect value during a test of several Shift instructions. The test, made within the arithmetic logic portion of the CPU test, utilizes R2 to perform a series of Shift Right Arithmetic, Shift Right Logical, and Shift Right Rotate (SRA, SRL and SRR) instructions. The result of these operations is compared to the value contained in R5. If the contents of R2 and R5 are not equal, the self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
BHE/BL INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 7 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=2222 0000 R4=3333 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Compare Register, Branch If Higher Or Equal and the Branch If Less (CR, BHE and BL) instructions. The test performs a compare between the contents of General Purpose Registers 2 and 4 (R2 and R4). If R2 < R4, as intended, proper execution of the BHE and BL instructions should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '11'), or 3) R2 or R4 contains an incorrect value, such that R2 > R4. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
BLE INSTRUCTION FAILED TO BRANCH **ERROR CODE: 7 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=2222 0000 R4=3333 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch If Less Or Equal (BLE) instruction. The test performs a compare between the contents of General Purpose Registers 2 and 4 (R2 and R4). If R2 < R4, as intended, proper execution of the BLE instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '11'), or 3) R2 or R4 contains an incorrect value, such that R2 > R4. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
BLE/BH INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 7 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=2222 0000 R4=3333 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch If Less Or Equal and the Branch If High (BLE and BH) instructions. The test performs a compare between the contents of General Purpose Registers 2 and 4 (R2 and R4). If R4 > R2, as intended, proper execution of the BLE and BH instructions should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '01'), or 3) R2 or R4 contains an incorrect value, such that R2 > R4. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
BHE INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 7 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=2222 0000 R4=3333 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch If High Or Equal (BHE) instruction. The test performs a compare between the contents of General Purpose Registers 2 and 4 (R2 and R4). If R4 > R2, as intended, proper execution of the BHE instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '01'), or 3) R2 or R4 contains an incorrect value, such that R2 > R4. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
BM INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 7 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3 =0004 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch If Mixed (BM) instruction. The test performs an And Halfword Immediate (NHI) instruction using the contents of General Purpose Register 3 (R3) and 'FFFF'. If the result contains both 1's and 0's, as intended, proper execution of the BM instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '11'), or 3) R3 contains an incorrect value, such that the AND operation results in all 0's. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
BE INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 7 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=7F33 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch If Equal (BE) instruction. The test performs a compare between the most significant 16 bits of General Purpose Register 3 (R3) and '7F33'. If the halfword equals '7F33', as intended, proper execution of the BE instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) R3 contains a value other than '7F33' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
TRB/BO INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 7 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=7F33 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Test Register Bits and Branch If Ones (TRB and BO) instructions. The test performs a TRB instruction on the most significant 16 bits of General Purpose Register 3 (R3), using the value '7F33'. If every bit position tested equals "1", as intended, proper execution of the BO instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '01'), or 3) R3 contains a value other than '7F33' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
OHI INSTRUCTION FAILURE **ERROR CODE: 7 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=FFFF 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Or Halfword Immediate (OHI) instruction. The test performs an OHI instruction using the most significant 16 bits of General Purpose Register 3 (R3) and the value '80CC'. If the result equals 'FFFF', as intended, the error should not occur to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) R3 contains a value other than '7F33' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
OVERFLOW NOT SET OR BOV INSTRUCTION FAILURE **ERROR CODE: 7 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0001 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Add Halfword Immediate and Branch On Overflow (AHI and BOV) instructions. The test adds '0002' to the most significant 16 bits of General Purpose Register 3 (R3). If the result equals '0001' with an overflow condition set, as intended, proper execution of the BOV instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Overflow Indicator in the Program Status Word (PSW, bit 19) was not correctly set to "1", or 3) R3 contains a value other than 'FFFF' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
CARRY BIT NOT SET OR BOC INSTRUCTION FAILURE **ERROR CODE: 7 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0001 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch On Carry (BOC) instruction. The test adds '0002' to the most significant 16 bits of General Purpose Register 3 (R3). If the result equals '0001' with a carry condition set, as intended, proper execution of the BOC instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Carry Indicator in the Program Status Word (PSW, bit 18) was not correctly set to "1", or 3) R3 contains a value other than 'FFFF' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
SIGN BIT NOT SET OR BP INSTRUCTION FAILURE **ERROR CODE: 8 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0001 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch If Positive (BP) instruction. The test adds '0002' to the most significant 16 bits of General Purpose Register 3 (R3). If the result equals '0001', as intended, proper execution of the BP instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '01'), or 3) R3 contains a value other than 'FFFF' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
LACR INSTRCTN DID NOT SET CC TO INDICATE LT 0 **ERROR CODE: 8 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=FFFF 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Load Arithmetic Complement, Branch Not Minus and Branch Not Zero (LACR, BNM and BNZ) instructions. The test performs a LACR instruction, which places the two's complement of General Purpose Register 3 (R3) back into R3. If the result equals 'FFFF', as intended, proper execution of the BNM and BNZ instructions should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '11'), or 3) R3 contains a value other than '0001' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
AH/BP/BZ INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 8 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0000 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Add Halfword, Branch If Positive and Branch If Zero (AH, BP and BZ) instructions. The test adds '0001' to the most significant 16 bits of General Purpose Register 3 (R3). If the result equals '0000', as intended, proper execution of the BP and BZ instructions should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) R3 contains a value other than '0000' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
TS -- TEST AND SET -- INSTRUCTION FAILURE **ERROR CODE: 8 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a check of the Test and Set (TS) instruction. The instruction tests a halfword memory location specified by the contents of General Purpose Register 7 (R7) for a value '0000', and if true, sets the Condition Code to '00' (Program Status Word bits 16-17) and the memory location to 'FFFF'. Proper execution of the Branch On Zero instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the CC was not correctly set (s/b '00'), or 3) the contents of R7 contains a value other than the intended test location. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
TB/BO INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 8 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Test Bits and Branch If Ones (TB and BO) instructions. The TB instruction tests a halfword memory location specified by the contents of General Purpose Register 2 (R2) for the value of 'FFFF', and if true, sets the Condition Code to '01' (Program Status Word bits 16-17). Proper execution of the BO instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '01'), or 3) the contents of R2 contains a value other than the intended test location. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
BVC INSTRUCTION OR CONDITION CODE FAILURE **ERROR CODE: 8 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=C3FF 421F.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch On Overflow and Carry (BVC) instruction. The test subtracts '3C00 BDE1' from General Purpose Register 4 (R4), which contains '0000 0000'. This operation should result in a Carry condition and proper execution of the BVC instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Carry Indicator in the Program Status Word (PSW, bit 18) was not correctly set to "1", or 3) R4 contains a value greater than '3C00 BDE1', such that the Carry Indicator would not be set. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
DR/MHI/CHI INSTRUCTION OR COND CODE FAILURE **ERROR CODE: 8 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=1649 0000 R5=F421 F3C0.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Divide Register, Multiply Halfword and Compare Halfword (DR, MHI and CHI) instructions. The test performs a series of multiply and divide instructions using General Purpose Registers 4 and 5 (R4 and R5). If R4 equals '1649 0000', as intended, proper execution of the Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) R4 contains a value other than '1649' in its high order bits. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a General Purpose Register.
FLOATING POINT TEST 1 FAILURE **ERROR CODE: 8 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: F2=0000 0000 F3=EF1D DDDD F6=BE76 5432 F7=1000 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the floating point Compare (CE) instruction. The test verifies that two floating point numbers, consisting of a sign bit (bit 0), a characteristic or exponent (bits 1-7) and a fraction (bits 8-31), will successfully compare even though the exponents differ. If a floating point number contains a fraction of zero, as in Floating Point Registers 2 and 7 (F2 and F7), the Compare will ignore the sign and exponent, and a Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) F2 or F7 contains a value other than expected. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
FLOATING POINT TEST 2 FAILURE **ERROR CODE: 8 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: F4=0000 0000 F5=0000 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the floating point Load, Add and Subtract (LE, AE and SE) instructions, using both 32 and 64-bit operands. The test verifies that an even-odd pair of Floating Point Registers (F4 and F5) can be referenced as one 64-bit operand, by subtracting a 64-bit constant and verifying a result of zero. A Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) F4 and F5 contain a value other than '4412 3431 BA66 6541' prior to a Subtract instruction. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.
FLOATING POINT TEST 3 FAILURE **ERROR CODE: 8 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: F3=EF1D DDDD.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the floating point Add (AE) instruction. The test verifies that a Floating Point Register (F3) can be added to the contents of a memory location specified by direct addressing, both of which are equal. If F3 equals 'EF1D DDDD', as intended, proper execution of the Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) F3 or the memory location contain a value other than 'EEEE EEEE' prior to an Add instruction. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
FLOATING POINT TEST 4 FAILURE **ERROR CODE: 9 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: F3=0000 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the floating point Add (AE) instruction. The test verifies that a floating point number, consisting of a sign bit (bit 0), a characteristic or exponent (bits 1-7) and a fraction (bits 8-31), will be set to true zero when an operation causes exponent underflow to occur. If Floating Point Register 2 (F2) is set to '0000 0000', a Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) F2 or the memory location contains a value other than expected. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.
FLOATING POINT TEST 5 FAILURE **ERROR CODE: 9 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: F2 and F3=0000 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the floating point Load, Multiply and Subtract (LED, MED and SED) instructions, using 64-bit operands. The test verifies that an even-odd pair of Floating Point Registers (F2 and F3) can be referenced as one 64-bit operand, by performing a series of arithmetic operations. A Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) F2 and F3 contain a value other than '0000 0000' prior to a Subtract instruction. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
FLOATING POINT TEST 6 FAILURE **ERROR CODE: 9 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: F0 and F1=0000 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the floating point Load and Divide (LED and DED) instructions, using 64-bit operands. The test verifies that an even-odd pair of Floating Point Registers (F2 and F3) can be referenced as one 64-bit operand, by performing a series of arithmetic operations. A Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) F0 and F1 contain a value other than '4146 2D5A 2910 5916' prior to a Subtract instruction. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.
BRANCH INDIRECT FAILED **ERROR CODE: 9 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=0000 0000 R7=0003 0000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Branch Indirect (BC) instruction. The test verifies that the program branches to an address formed by adding the contents of General Purpose Register 6 (R6) to the contents of a memory location. If R6 equals '0000 0000', as intended, proper execution of the Branch instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 2 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) R6 or the memory location contain a value other expected. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
SRDR SHIFT FAILED **ERROR CODE: 9 4**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=0000 6000.	An unexpected Supervisor Call (SVC) instruction, used by the self-test program to annunciate errors, has been executed during a test of the Shift Right Double and Rotate (SRDR) instruction. The test verifies that an even-odd pair of General Purpose Registers (R6 and R7) can be referenced as one 64-bit operand for a shift operation. The SRDR instruction will shift bits from the low order of one register to the high order of the other register, so that no bit are lost. A Branch If Equal instruction should not cause the error to be annunciated. In order for the error to be annunciated, any of 3 conditions must be true during the test: 1) the instruction logic did not execute properly, 2) the Condition Code (CC) in the Program Status Word (PSW, bits 16-17) was not correctly set (s/b '00'), or 3) R6 or R7 contains values other than '0000 0000' and '0003 0000', respectively, prior to the SRDR instruction. The self-test program will annunciate the error and continue.	Failure of the CPU, the CPU Read Only Storage (ROS) or a Floating Point Register.
ADDRESSING INTRPT FAILURE ON ILLEGAL ADDRESS **ERROR CODE: 9 5**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=FFFF 0000.	The CPU did not generate a Program Check (PC) interrupt when an access of an out-of-bounds memory location was attempted. The self-test program verifies the interrupt logic by setting the Data Sector Register (DSR) in the current Program Status Word (PSW) to its maximum value using a Load Program Status (LPS) instruction, and loading General Purpose Register 4 (R4) with 'FFFF 0000'. An attempt is then made to load R4 with the halfword addressed by the DSR and the contents of R4. Since the address specified is invalid, the CPU should generate a PC CPU address specification interrupt with an interrupt code of '0003', and store it along with the current PSW into the PC PSW low core locations 48-4B. The self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
STORE-PROTECT INTERRUPT FAILED TO OCCUR **ERROR CODE: 9 6**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R5=0000 0000.	The CPU did not generate a Program Check (PC) interrupt when a write to a protected memory location was attempted. The self-test program verifies the interrupt logic by setting General Purpose Register 5 (R5) to '0000 0000', then attempting to store the most significant halfword (bits 0-15) of this register into a protected halfword specified by a program constant. Since the location is store protected, the CPU should generate a PC CPU store protect interrupt with an interrupt code of '0007', and store it along with the current PSW into the PC PSW low core locations 48-4B. The self-test program will annunciate the error, set the store-protect bit of the test location, reset the interrupt return address and continue.	Failure of the CPU or incorrect store protect bit setting in the test location.
STORE-PROTECT INTERRUPT CODE NOT 7 **ERROR CODE: 9 7**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=0007 0000.	The CPU has generated a Program Check (PC) store protect interrupt with an incorrect interrupt code in the Program Status Word (PSW). The interrupt code is generated by the CPU following detection of the store protect violation, and is contained in the PC PSW (low core loc. 48-4B). The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the PC interrupt handler routine, into General Purpose Register 7 (R7). A check is then made to determine if the returned value is equal to the known value of '0007'. If not, the self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU.
INT CODE NOT HEX F AFTER INS MON INTERRUPT **ERROR CODE: 9 8**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=000F 0000.	The CPU has generated an Instruction Monitor (IM) interrupt with an incorrect interrupt code in the Program Status Word (PSW). The interrupt code is generated by the CPU following an attempt to execute data stored in an unprotected memory location as program code, and is contained in the IM PSW (loc. 70-73). The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the IM interrupt handler routine, into General Purpose Register 7 (R7). A check is then made to determine if the returned value is equal to the known value of '000F'. If not, the self-test program will annunciate the error and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
FIXED POINT OVERFLOW NOT MASKABLE **ERROR CODE: 9 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=FFFE 0000.	The CPU generated a Program Check (PC) fixed point overflow interrupt while the proper mask was set in the current Program Status Word (PSW). A fixed point overflow interrupt is generated by the CPU when the result of an arithmetic operation is too large to be stored in the specified format. The self-test program verifies the interrupt logic by executing a Set Program Mask (SPM) instruction using the value of '11EA' and adding the highest positive halfword value ('7FFF') to itself. Bits 16-23 of the value '11EA' replace bits 16-23 of the current PSW, thus setting the fixed point overflow mask (bit 20) to '0'. This should inhibit the annunciation of any fixed point overflow condition. If the overflow interrupt occurs following execution of the Add operation, the CPU will generate an interrupt code of '0004', and store it along with the current PSW into the PC PSW low core locations 48-4B. The self-test program will annunciate the error and continue.	Failure of the CPU or the current PSW.
FIXED POINT OVERFLOW INTERRUPT FAILED TO OCCUR **ERROR CODE: 1 0 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=PC interrupt return address R7=FFFE 0000.	The CPU did not generate a Program Check (PC) fixed point overflow interrupt following an arithmetic operation which resulted in a value too large to be stored in the specified format. The self-test program verifies the interrupt logic by executing a Set Program Mask (SPM) instruction using the value of '0B00' and adding the highest positive halfword value ('7FFF') to itself. Bits 16-23 of the value '0B00' replace bits 16-23 of the current PSW, thus setting the fixed point overflow mask (bit 20) to '1'. This will enable the annunciation of any fixed point overflow condition. Upon detection of the overflow condition, the CPU should generate a PC interrupt with an interrupt code of '0004', and store it along with the current PSW into the PC PSW low core locations 48-4B. The self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
INT CODE NOT 4 AFTER FIXED PT OVFLO INTERRUPT **ERROR CODE: 101**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=0004 0000.	The CPU has generated a Program Check (PC) fixed point overflow interrupt with an incorrect interrupt code in the Program Status Word (PSW). The interrupt code is generated by the CPU following detection of the overflow condition, and is contained in the PC PSW (loc. 48-4B). The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the PC interrupt handler routine, into General Purpose Register 7 (R7). A check is then made to determine if the returned value is equal to the known value of '0004'. If not, the self-test program will annunciate the error and continue.	Failure of the CPU.
MCH CHK INTRPT NOT MASKABLE WITH CPU MEM PARITY **ERROR CODE: 102**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=4300 1234.	The CPU generated a Machine Check (MC) CPU memory parity interrupt while the proper mask was set in the current Program Status Word (PSW). A memory parity interrupt is generated by the CPU whenever data containing bad (even) parity is retrieved from CPU memory (loc. 00000-27FFF). The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of '0008' and attempting to load into General Purpose Register 4 (R4) the fullword of data contained at locations 2 and 3. The value '0008' replaces bits 32-47 of the current PSW, thus setting the MC mask (bit 45) to '0'. This will inhibit the generation of all interrupts. If the memory parity interrupt occurs following execution of the load operation, the CPU will generate an interrupt code of '0003' and store it along with the current PSW into low core locations 40-43. The self-test program will annunciate the error, reset the interrupt return address, reset the fail indication and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MCH CHK INTRPT DID NOT OCCUR ON CPU MEM PARITY **ERROR CODE: 103**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=4300 1234.	The CPU did not generate a Machine Check (MC) CPU memory parity interrupt following an attempt to retrieve a fullword of data containing bad (even) parity from CPU memory. A MC interrupt should be generated by the CPU whenever even parity is detected on data read from memory. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'FF0C' and attempting to load into General Purpose Register 4 (R4) the fullword of data contained at locations 2 and 3. The value 'FF0C' replaces bits 32-47 of the current PSW, thus setting the MC mask (bit 45) to '1'. This will enable the generation of all interrupts. If the MC interrupt does not occur following execution of the load instruction, the self-test program will annunciate the error, reset the interrupt return address, reset the fail indication and continue.	Failure of the CPU.
MCH CHK INT CODE NOT 3 AFTER CPU MEMORY PARITY **ERROR CODE: 104**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=0003 0000.	The CPU has generated a Machine Check (MC) CPU memory parity interrupt with an incorrect interrupt code in the Program Status Word (PSW). The interrupt code is generated by the CPU following the detection of even parity in a memory read from CPU memory, and is contained in the MC PSW (loc. 40-43). The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the MC interrupt handler routine, into General Purpose Register 7 (R7). A check is then made to determine if the returned value is equal to the known value of '0003'. If not, the self-test program will annunciate the error and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
R/M VOTER TEST 1 FAILURE **ERROR CODE: 105**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0010 0000 R4=8808 0000 R5=0000 0000 R6=401E 0000 R7=8810 0000.	A test of the Redundancy Management (R/M) Voter logic has failed. The test utilizes a Program Controlled Output (PCO) instruction to load the Voter Test Register, which sets failure votes against the tested GPC, one at a time. The program reads the R/M Status Register to verify that the GPC did not set any of its fail latches, which should only be set when 2 or more fail votes are annunciated against it. The self-test program will annunciate the error, reset the failure indications and continue. R/M Status Register format: (0-no failure, 1-failure) Bit 0=Fail/Timeout latch Bit 1,2=Inhibit fail vote inputs and outputs during self-test Bits 3-6=Failure vote inputs 1-4 Bits 7-10=Failure vote outputs 1-4 Bits 11-14=self-test failure votes 1-4 Bits 15,16=Voter Fail and Timeout Latches Bits 17,18=Voter and Timer Termination Control latches Bit 19=IOP Transmission Termination Bits 20-31=Watchdog Timer value. Voter Test Register format: Bits 0-26=Undefined Bit 27=Voter Test Control (0-normal operations, 1-perform test) Bits 28-31=Voter test inputs 1-4.	Failure of the IOP, R/M Voting logic or the R/M Status Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
COULD NOT SET TIME OUT LATCH WITH WATCHDOG **ERROR CODE: 106**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=0000 0FFF, R3=0814 0000, R4=8804 0000, R5=WD timer pre-test value, R6=0000 0000.	A test of the Watchdog (WD) Timer logic has failed. The test utilizes a Program Controlled Output (PCO) instruction to load the timer with 'FFF', which causes it to timeout on the next count (.768 msec). The program then reads the Redundancy Management (R/M) Status Register to verify that the value is equal to zero following the WD timeout interrupt, since the timer will not operate once a timeout has occurred, until reset. If bits 20-31 of the R/M Status Register are not all zero, the self-test program will annunciate the error and continue. R/M Status Register format: (0-no failure, 1-failure) Bit 0=Fail/Timeout Latch Bit 1=Inhibit fail vote inputs during self-test Bit 2=Inhibit fail vote outputs during self-test Bits 3-6=Failure vote inputs 1-4 Bits 7-10=Failure vote outputs 1-4 Bits 11-14=self-test failure votes 1-4 Bit 15=Voter Fail Latch Bit 16=Timeout Latch Bit 17=Voter Termination Control Latch Bit 18=Timer Termination Control Latch Bit 19=IOP Transmission Termination Bits 20-31=Watchdog Timer value.	Failure of the IOP, the Watchdog Timer logic or the R/M Status Register.
DID NOT GET AN EXPECTED EXT 0 INTERRUPT **ERROR CODE: 107**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=EX0 interrupt return address.	The IOP did not generate an expected External 0 (EX0) Watchdog Timer interrupt following a test of the Watchdog timer logic. The test issues a Program Controlled Output (PCO) instruction to the IOP, while all External interrupts are masked, to load the timer with 'FFF'. This will cause the Watchdog Timer to expire on the next count (.768 msec). An EX0 interrupt should be generated whenever the timer expires before being reset. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'FF0C', which allows all pending External interrupts to be annunciated. The value 'FF0C' replaces bits 32-47 of the current PSW, thus setting the EX0 mask (bit 35) to '1'. If the EX0 interrupt does not occur following execution of the SSM instruction, the self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the IOP or the IOP interrupt logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
WATCHDOG INT BIT NOT ON AFTER EXT 0 INTERRUPT **ERROR CODE: 108**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=INTA.	The IOP generated an External 0 (EX0) Watchdog Timer interrupt without setting bit 0 in Interrupt Register A (INTA) to '1', which indicates that the Watchdog (Go/No-Go) timer has expired. The contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX0 PSW (loc. 78-7B). The self-test program verifies the interrupt logic by loading the value of the interrupt code (INTA), returned from the EX0 interrupt handler routine, into General Purpose Register 4 (R4). A check is then made to determine if the returned value is equal to the known value of '8000'. If not, the self-test program will annunciate the error and continue.	Failure of the IOP, the IOP interrupt logic or Interrupt Register A.
EXPECTED EXT 4 INTERRUPT DID NOT OCCUR **ERROR CODE: 109**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The IOP did not generate an expected External 4 (EX4) interrupt following a test of the IOP interrupt logic. The test issues a Test Register Program Controlled Output (PCO) instruction to the IOP while all External interrupts are masked, which causes all External 0,1,3 and 4 interrupts to be generated. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'FF0C', which allows all pending External interrupts to be annunciated. The value 'FF0C' replaces bits 32-47 of the current PSW, thus setting the EX4 mask (bit 39) to '1'. If no EX4 interrupts occur following execution of the SSM instruction, the self-test program will annunciate the error and continue.	Failure of the IOP or the Control/Monitor.
EXPECTED EXT 3 INTERRUPT DID NOT OCCUR **ERROR CODE: 110**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The IOP did not generate an expected External 3 (EX3) interrupt following a test of the IOP interrupt logic. The test issues a Test Register Program Controlled Output (PCO) instruction to the IOP while all External interrupts are masked, which causes all External 0,1,3 and 4 interrupts to be generated. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'E20C', which allows all pending External 3 interrupts to be annunciated. The value 'E20C' replaces bits 32-47 of the current PSW, thus setting the EX3 mask (bit 38) to '1'. If no EX3 interrupts occur following execution of the SSM instruction, the self-test program will annunciate the error and continue.	Failure of the IOP or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
EXPECTED EXT 1 INTERRUPT DID NOT OCCUR **ERROR CODE: 111**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The IOP did not generate an expected External 1 (EX1) interrupt following a test of the IOP interrupt logic. The test issues a Test Register Program Controlled Output (PCO) instruction to the IOP while all External interrupts are masked, which causes all External 0,1,3 and 4 interrupts to be generated. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'E80C', which allows all pending External 1 interrupts to be annunciated. The value 'E80C' replaces bits 32-47 of the current PSW, thus setting the EX1 mask (bit 36) to '1'. If no EX1 interrupts occur following execution of the SSM instruction, the self-test program will annunciate the error and continue.	Failure of the IOP or the Control/Monitor logic.
EXPECTED EXT 0 INTERRUPT DID NOT OCCUR **ERROR CODE: 112**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The IOP did not generate an expected External 0 (EX0) interrupt following a test of the IOP interrupt logic. The test issues a Test Register Program Controlled Output (PCO) instruction to the IOP while all External interrupts are masked, which causes all External 0,1,3 and 4 interrupts to be generated. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'F00C', which allows all pending External 0 interrupts to be annunciated. The value 'F00C' replaces bits 32-47 of the current PSW, thus setting the EX0 mask (bit 35) to '1'. If no EX0 interrupts occur following execution of the SSM instruction, the self-test program will annunciate the error and continue.	Failure of the IOP or the Control/Monitor logic.
UNEXPECTED EXTRN 1 INTRPT- INTERRUPT CODE > 6 **ERROR CODE: 113**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G4=memory location of the EX1 PSW	The CPU has generated an unexpected External 1 (EX1) interrupt with an incorrect interrupt code contained in the MC Program Status Word (PSW). The interrupt code is generated by the CPU following detection of an error condition, and is contained in the EX1 PSW (loc. 80-83). The self-test program verifies the interrupt code by loading the value stored at low core location 83 into General Purpose Register 4 (G4). A check is then made to determine if the returned value is within the valid range of '0000' through '0006'. If not, the self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED INT REG E VALUE AFTER EXT 4 INTERRUPT **ERROR CODE: 114**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=INTE.	The IOP failed to set the significant bits in Interrupt Register E (INTE), following a test which forces all External 4 (EX4) interrupts to occur. The program issues a Test Register Program Controlled Output (PCO) instruction to the IOP, which causes all External 0,1,3 and 4 interrupts to be generated, then executes a Set System Mask (SSM) instruction to allow each class of interrupt verified separately. The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the EX4 interrupt handler routine, into General Purpose Register 6 (R6). A check is then made to determine if the retufned value is equal to the known value of 'F000 0000'. If not, the self-test program will annunciate the error and continue. The contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX4 PSW (loc. 98-9B). Interrupt Register E format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the IOP or the Control/Monitor logic.
UNEXPECTED INT REG D VALUE AFTER EXT 3 INTERRUPT **ERROR CODE: 115**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=INTD.	The IOP failed to set the significant bits in Interrupt Register D (INTD), following a test which forces all External 3 (EX3) interrupts to occur. The program issues a Test Register Program Controlled Output (PCO) instruction to the IOP, which causes all External 0,1,3 and 4 interrupts to be generated, then executes a Set System Mask (SSM) instruction to allow each class of interrupt verified separately. The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the EX3 interrupt handler routine, into General Purpose Register 6 (R6). A check is then made to determine if the returned value is equal to the known value of '8000 0000'. If not, the self-test program will annunciate the error and continue. The contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX3 PSW (loc. 90-93). Interrupt Register D format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the IOP or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED INT REG B VALUE AFTER EXT 1 INTERRUPT **ERROR CODE: 116**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=INTB.	The IOP failed to set the significant bits in Interrupt Register B (INTB), following a test which forces all External 1 (EX1) interrupts to occur. The program issues a Test Register Program Controlled Output (PCO) instruction to the IOP, which causes all External 0,1,3 and 4 interrupts to be generated, then executes a Set System Mask (SSM) instruction to allow each class of interrupt verified separately. The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the EX1 interrupt handler routine, into General Purpose Register 6 (R6). A check is then made to determine if the returned value is equal to the known value of 'EC00 0000'. If not, the self-test program will annunciate the error and continue. The contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX1 PSW (loc. 80-83). Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED INT REG A VALUE AFTER EXT 0 INTERRUPT **ERROR CODE: 117**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=INTA.	The IOP failed to set the significant bits in Interrupt Register A (INTA), following a test which forces all External 0 (EX0) interrupts to occur. The program issues a Test Register Program Controlled Output (PCO) instruction to the IOP, which causes all External 0,1,3 and 4 interrupts to be generated, then executes a Set System Mask (SSM) instruction to allow each class of interrupt verified separately. The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the EX0 interrupt handler routine, into General Purpose Register 6 (R6). A check is then made to determine if the returned value is equal to the known value of 'F800 0000'. If not, the self-test program will annunciate the error and continue. The contents of the current Program Status Word (PSW) at the time the interrupt occurred is stored in the EX0 PSW (loc. 78-7B). Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the IOP or the Control/Monitor logic.
NO EXPECTD EXT 1 INT AFTER BAD PRTY WRITE BY MSC **ERROR CODE: 118**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The CPU did not generate an expected External 1 (EX1) interrupt following a test of the CPU parity checker circuitry. The test issues a Program Controlled Output (PCO) instruction to the IOP to command it to generate bad (even) parity. The CPU will then enable all interrupts by executing a Set System Mask (SSM) instruction using the value of 'FF0C'. This value replaces bits 32-47 of the current PSW, thus setting the EX1 mask (bits 36 and 45) to '1'. An MSC program is then initiated which will attempt to write data to main memory using a Direct Memory Access (DMA) 'In' operation. Since the IOP was commanded to generate even parity, the data transferred to main memory for the write will contain bad parity. Upon detection of the parity error, the CPU should generate an EX1 DMA Data Parity Interrupt, with interrupt code '0002' and the current Program Status Word (PSW) stored into low core locations 80-83. If no interrupt occurs, the self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU Interrupt logic, the CPU parity checker circuitry or the IOP DMA parity generator.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
IOP REG NOT 0 AFTER EXT 1 BAD PARITY WRITE INT **ERROR CODE: 119**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=INTB.	The IOP has incorrectly set Interrupt Register B (INTB) to a non-zero value following an External 1 (EX1) Direct Memory Access (DMA) write parity interrupt. INTB is utilized to identify the External 1 interrupts which are generated by the IOP. The DMA data write parity interrupt is generated by the CPU, following the detection of bad (even) parity in data transferred from the IOP during a DMA 'In' (main memory write) operation. The self-test program verifies the interrupt logic by loading the value of INTB, returned from the EX1 interrupt handler routine, into General Purpose Register 4 (R4). A check is then made to determine if the returned value is equal to zero. If not, the self-test program will annunciate the error and continue. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP, the IOP Control/Monitor logic or INTB.
INT CODE NOT 1 AFTER EXT 1 PCI DATA PARITY INT **ERROR CODE: 120**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=0001 0000.	The CPU has generated an External 1 (EX1) Program Controlled Input (PCI) data parity interrupt with an incorrect interrupt code in the Program Status Word (PSW). An interrupt code of '0001' is generated by the CPU following the detection of even parity in data transferred from the IOP during a PCI operation, and is stored along with the current PSW into low core locations 80-83. The self-test program verifies the interrupt logic by loading the known value of the interrupt code into General Purpose Register 4 (R4). A check is then made to determine if the known value is equal to the contents of low core memory location '83'. If not, the self-test program will annunciate the error and continue.	Failure of the CPU interrupt logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
NO EXPECTED EXT 1 INT ON BAD DMA ADDRS PARITY **ERROR CODE: 121**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=C110 0000.	The CPU did not generate an expected External 1 (EX1) interrupt following a test of the CPU parity checker circuitry. The test issues a Program Controlled Output (PCO) instruction to the IOP to command it to generate bad (even) address parity. An MSC program is executed which will attempt to fetch an instruction from main memory using a Direct Memory Access (DMA) 'Out' operation. Since the IOP was commanded to generate even address parity, the address transferred to the CPU for the read will contain bad parity. Upon detection of the parity error, the CPU should generate an EX1 DMA Address Parity Interrupt, with interrupt code '0005' and the current Program Status Word (PSW) stored into low core locations 80-83. If no interrupt occurs, the self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU interrupt logic, the CPU parity checker circuitry or the IOP DMA parity generator.
INT CODE NOT 5 AFTER EXT 1 ADDRS PARITY INT **ERROR CODE: 122**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=0005 0000.	The CPU has generated an External 1 (EX1) Direct Memory Address (DMA) address parity interrupt with an incorrect interrupt code in the Program Status Word (PSW). An interrupt code of '0005' is generated by the CPU following the detection of even parity in an address transferred from the IOP during a DMA 'Out' operation (instruction fetch), and is stored along with the current PSW into low core locations 80-83. The self-test program verifies the interrupt logic by loading the known value of the interrupt code into General Purpose Register 4 (R4). A check is then made to determine if the known value is equal to the contents of low core memory location '83'. If not, the self-test program will annunciate the error and continue.	Failure of the CPU interrupt logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
IOP REG NOT 0 AFTER EXT 1 DMA ADDRESS PARITY INT **ERROR CODE: 123**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=INTB.	The IOP has incorrectly set Interrupt Register B (INTB) to a non-zero value following an External 1 (EX1) Direct Memory Access (DMA) address parity interrupt. INTB is utilized to identify the External 1 interrupts which are generated by the IOP. The DMA address parity interrupt is generated by the CPU following the detection of bad (even) parity in the address transferred from the IOP during a DMA 'Out' (instruction fetch) operation. The self-test program verifies the interrupt logic by loading the value of INTB, returned from the EX1 interrupt handler routine, into General Purpose Register 4 (R4). A check is then made to determine if the returned value is equal to zero. If not, the self-test program will annunciate the error and continue. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP, the Control/Monitor logic or INTB.
GPC POWER FAIL AT LOCATION SHOWN - RECOVERED OK **ERROR CODE: 124**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The GPC has successfully recovered from a power failure which interrupted normal processing. Upon detection of a power transient (i.e. voltage drops below 17.5), the CPU will perform a Power Down Putaway routine, which utilizes a 100 microsecond window to store each of the general register sets, the clock values and the contents of other program variables necessary to restore the computer to the configuration it was in prior to the failure. The next time the computer is powered on, or if the power failure was transient, control is passed to the Power Fail Executive routine using a branch through the Power On Program Status Word (PSW, low core location '0004'). The routine, made up of 12 separate modules, will determine where in the program the power failure occurred and what steps need to be taken to recover. The self-test program will only store the SVC number of this error (124) for later annunciation and continue with the recovery procedure.	Failure of the GPC 28V power supply.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MSC @STP OR BCE #STP WRAP FAILED **ERROR CODE: 125**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=0000 0000 R4=FFFF 0000.	The MSC or at least one BCE has encountered an error during execution of the processor self-test routines @STP and #STP. Each routine verifies the proper function of the processors using built-in test instructions. Detection of any error during the testing will cause bit 22 of the appropriate processor status register to be set to '1' and the proceoosr wil be set to No-Go. The MSC's addressing logic, data flow, arithmetic logic, parity detection circuitry, channel operation and failure vote logic are tested using the @STP instruction. The BCE's addressing logic, data flow, Direct Memory Access (DMA) logic, channel queues and MIA circuitry are tested using the #STP instruction. Part of the self-test requires each processor to read the contents of a low core memory location, then write the same value into a second pre-determined location. Following completion of the tests, the CPU verifies the contents of these memory locations, along with the value of each status register. If any error is detected, a flag is set to 'FFFF 0000' indicate an error has occurred, and the respective bit in a string will be zeroed to remove the failed processor from further testing. The self-test program will annunciate the error and continue.	Failure of the MSC or a BCE.
MSC NOT IN WAIT AFTER BAD ADDRESS PARITY TEST **ERROR CODE: 126**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=FFFF FF80 R5=Go/No-Go register R6=1000 0000.	An External 1 (EX1) Direct Memory Access (DMA) address parity interrupt, which was intentionally caused during verification of the CPU parity checker logic, has failed to place the MSC into a wait state. During the IOP test routine, the CPU disables all EX1 interrupts using a Set System Mask (SSM) instruction, then reads the Status 1 (Go/No-Go) Register using a Program Controlled Input (PCI) instruction. Since the parity error was forced during an instruction fetch from main memory, an illegal op code will be encountered which sets the MSC into a wait state (bit 0 of the Go/No-Go register equal to '0'). If the bit is equal to '1' (Go), the self-test program will annunciate the error and continue.	Failure of the IOP, the IOP Control/Monitor logic, the Go/No-Go register or the MSC.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
NO EXPECTED MCH CHK IOP STORAGE PARITY INTERRUPT **ERROR CODE: 127**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=9204 0000 R7=A201 0000.	The CPU did not generate an expected Machine Check (MC) interrupt during a test of the CPU parity checker circuitry. The test issues a Program Controlled Output (PCO) instruction to the IOP to load the MSC program counter with the address of a program which will read a main memory location which contains bad parity. The MSC program is then initiated using a second PCO instruction and an attempt is made to read a fullword of data from location '2' of main memory. Upon detection of the parity error, the CPU should generate a MC IOP Storage Parity Interrupt, with interrupt code '0002' and the current Program Status Word (PSW) stored into low core locations 40-43. If no interrupt occurs, the self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU Interrupt logic or parity checker logic.
INT CODE NOT 2 AFTER MCH CHK IOP STRGE PRITY INT **ERROR CODE: 128**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R7=0002 0000.	The CPU has generated a Machine Check (MC) IOP storage parity interrupt with an incorrect interrupt code in the Program Status Word (PSW). An interrupt code of '0002' is generated by the CPU following the detection of even parity in a fullword read from main memory location '2' and is stored along with the current PSW into low core locations 40-43. The self-test program verifies the interrupt logic by loading the value of the interrupt code, returned from the MC interrupt handler routine, into General Purpose Register 7 (R7). A check is then made to determine if the returned value is equal to the known value of '0002'. If not, the self-test program will annunciate the error, reset the CPU fault indicator and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
REAL TIME MSC TIME OUT **ERROR CODE: 1 2 9**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G2=0808 0000 G3=INTC.	An unexpected External 2 (EX2) Interrupt has been generated and the contents of Interrupt Register C (INTC) indicates that the MSC has timed out while waiting for all BCE's to return to the Wait state. The error can occur during either a Display Electronics Unit (DEU) load, Mass Memory Unit (MMU) I/O or downlist processing, and indicates that a BCE is hung 'busy'. Upon detection of the error, the MSC program will set INTC to '1000 0000' to indicate an MSC timeout, deschedule the DEU, MMU or D/L job and execute the Interrupt (@INT) instruction to interrupt the CPU. The CPU will then generate the EX2 interrupt, store the current Program Status Word (PSW) into low core locations 88-8B and execute the EX2 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of a DEU, BCE, MMU or PCMMU.
NO DEU POLL RESPONSE **ERROR CODE: 1 3 0**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The Display Electronics Unit (DEU) has not responded to a poll request by the GPC self-test program within a specified time limit. The DEU is polled two times per second, and its response is checked on the minor cycles immediately following those in which it is polled (7 and 20). A flag is set whenever a response has been received by the GPC. The self-test program will annunciate the error and return from the poll request routine to the job scheduler.	Failure of the DEU hardware or execution of the DEU Standalone Self Test Program (SMS D021 and SMS G099).
REAL TIME IPL RETRY FAILURE **ERROR CODE: 1 3 1**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G2=0808 0000 G3=INTC.	An unexpected External 2 (EX2) Interrupt has been generated and the contents of Interrupt Register C (INTC) indicates that a Display Electronics Unit (DEU) BITE status error has been detected on two consecutive attempts to perform a DEU load. The MSC will attempt the load twice before terminating the job. Upon detection of the timeout, the MSC program will check to see if the number of retries equals '2'. If not, the load is re-attempted. If the number of retries equals '2', INTC is set to '3000 0000' to indicate a BITE status error, the DEU load is descheduled and the Interrupt (@INT) instruction is executed to interrupt the CPU. The CPU will then generate the EX2 interrupt, store the current Program Status Word (PSW) into low core locations 88-8B and execute the EX2 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of a DEU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
DEU INITIALIZATION ERROR **ERROR CODE: 132**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G2=0808 0000 G3=INTC.	An unexpected External 2 (EX2) Interrupt has been generated and the contents of Interrupt Register C (INTC) indicates that a checksum error has been detected following a Display Electronics Unit (DEU) load. Upon completion of the load, the MSC program will check to see if bit 11 of the Software Status Word (SWSW) equal '0'. If so, the program continues. If the bit equals '1', the DEU load is descheduled to remove the CPU from the Wait state, INTC is set to '4000 0000' to indicate a DEU load failure and the Interrupt (@INT) instruction is executed to interrupt the CPU. The CPU will then generate the EX2 interrupt, store the current Program Status Word (PSW) into low core locations 88-8B and execute the EX2 interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the DEU load.
UNEXPECTED EXT 0 INTRPT: CODE = 0 /WATCHDOG TIMER/ **ERROR CODE: 133**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G3=INTA G4=memory location of the EX0 PSW G7=0800 0000.	An unexpected External 0 (EX0) Interrupt has been generated with the Watchdog (Go/No-Go) Timer bit set in Interrupt Register A (INTA), indicating a cyclic failure of the computer. The Go/No-Go timer must be periodically set by the CPU, using Program Controlled Output (PCO) instructions, to function properly. The IOP, upon detection that the Go/No-Go Timer has counted to its maximum value, will set INTA bit 0 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU or IOP, the C/M logic or Redundancy Management (R/M) logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXT 0 INTRPT: CODE = 0 /IOP FAIL LATCH/ **ERROR CODE: 134**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G3=INTA G4=memory location of the EX0 PSW G7=0800 0000.	An unexpected External 0 (EX0) Interrupt has been generated with the Voter Fail Latch bit set in Interrupt Register A (INTA), indicating a critical failure of the computer. The IOP voter logic receives the outputs of the fail vote discrete receivers and sets the voter fail latch whenever 2 or more fail votes are received from other GPCs. The IOP will set INTA bit 1 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or the R/M logic.
UNEXPCTD EXT 0 INTRPT: CODE = 0 CNTRL/MONITR IDLE **ERROR CODE: 135**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G3=INTA G4=memory location of the EX0 PSW G7=0800 0000.	An unexpected External 0 (EX0) Interrupt has been generated with the Control/Monitor (C/M) Idle bit set in Interrupt Register A (INTA), indicating the IOP C/M logic is available for operations. The C/M logic, which is controlled by the CPU using Program Controlled Output (PCO) instructions, controls and monitors such functions of the IOP as MIA transmit and receive enable, IOP discrete output, BCE halt/proceed status, interrupt register status and interruption of the CPU. The C/M Idle interrupt should only occur when a Master Reset IOP or IOP Fail Reset has been performed or at GPC power on. The IOP will set INTA bit 2 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP or C/M logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXPECTED EXT 0 INTRP: CODE = 0 IOP ROS PRITY ERROR **ERROR CODE: 136**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G3=INTA G4=memory location of the EX0 PSW G7=0800 0000.	An unexpected External 0 (EX0) Interrupt has been generated with the IOP Read Only Storage (ROS) parity error bit set in Interrupt Register A (INTA), indicating that a parity error has been detected during an attempt by the IOP to execute a microinstruction. The microinstruction, read from ROS and placed in the ROS data register (ROSDR), contained bad (even) parity when output from the ROSDR. Upon detection of the parity error, the IOP will set INTA bit 3 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU, the IOP, IOP PROM or the ROS data register.
UNEXPECTED EXT 0 INTRPT: CODE = 0 /IOP FAULT/ **ERROR CODE: 137**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G3=INTA G4=memory location of the EX0 PSW G7=0800 0000.	An unexpected External 0 (EX0) Interrupt has been generated with the IOP Fault bit set in Interrupt Register A (INTA), indicating that the IOP oscillator has stopped. The oscillator, contained in the Processor Element logic of the IOP, provides the timing necessary for the proper time-sharing of MSC and BCE functions. Upon detection that the oscillator has stopped, the IOP will set INTA bit 4 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the IOP, the Processor Element logic or the IOP oscillator.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
R/M VOTER TEST 2 FAILURE **ERROR CODE: 138**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=C001 0000 R7=8810 0000.	A test of the Redundancy Management (R/M) Voter logic has failed. The test uses a Program Controlled Output (PCO) instruction to load the Voter Test Register, which sets failure votes against the tested GPC two at a time. The program reads the R/M Status Register to verify that the GPC has set the appropriate fail latches, which should be set whenever 2 or more fail votes are annunciated against it. The self-test program will annunciate the error, reset the failure indications and continue. R/M Status Register format: (0-no failure, 1-failure) Bit 0=Fail/Timeout latch Bits 1-2=Inhibit fail vote inputs and outputs during self-test Bits 3-6=Failure vote inputs 1-4 Bits 7-10=Failure vote outputs 1-4 Bits 11-14=self-test failure votes 1-4 Bits 15,16=Voter Fail and Timeout Latches Bits 17,18=Voter and Timer Termination Control latches Bit 19=IOP Transmission Termination Bits 20-31=Watchdog Timer value. Voter Test Register format: Bits 0-26=Undefined Bit 27=Voter Test Control (0-normal operations, 1-perform test) Bits 28-31=Voter test inputs 1-4.	Failure of the IOP, R/M Voting logic or the R/M Status Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
I/O STORE PROTECT TEST FAILED **ERROR CODE: 139**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R2=MSC start address 2, 3 and 4: R3=interrupt code/ Interrupt Reg B R6=MSC program counter.	A test of the Direct Memory Access (DMA) store protect interrupt logic has failed. The test protects the fullword at memory location 2, then initiates an MSC program which attempts to write data into the same location. This should cause the CPU to generate an External 1 (EX1) interrupt, the MSC to enter a wait state and an interrupt code of '0004' to be stored into low core locations 80-83 with the current Program Status Word (PSW). The error can be annunciated for any of the following 4 reasons: 1. the EX1 interrupt never occurred, 2. the MSC did not go into a wait state following the interrupt, 3. the MSC was in a wait state at an incorrect memory location, or 4. the interrupt code generated by the CPU was not equal to '0004'. The program responses for 1-4 above are as follows: 1. annunciate the error, reset the interrupt return address, set bit 15 of General Register 5 (R5) to '1' and continue, 2. annunciate the error, halt the MSC, set bit 14 of R5 to '1' and continue, 3. annunciate the error, set bit 13 of R5 to '1' and continue, 4. annunciate the error, set bit 12 of R5 to '1' and continue.	Failure of the CPU or IOP, the IOP Control/Monitor logic or the MSC.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
COULD NOT RESET ALL TERMINATE CONTROL LATCHES **ERROR CODE: 140**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=0000 0000, R3=0814 0000, R4=8808 0000, R5=R/M Status register.	The CPU was unable to set both the Timer and Voter Termination Control Latches (TTCL and VTCL). The self-test program issues a Configure Control Latch Program Controlled Output (PCO) instruction to set both latches to '1', then reads the Redundancy/Management (R/M) Status Register to verify that both bits 17 and 18 are set. If not, the self-test program will annunciate the error, reset the latches, enable all processors and continue. R/M Status Register format: (0-no failure, 1-failure) Bit 0=Fail/Timeout Latch Bit 1=Inhibit fail vote inputs during self-test Bit 2=Inhibit fail vote outputs during self-test Bits 3-6=Failure vote inputs 1-4 Bits 7-10=Failure vote outputs 1-4 Bits 11-14=self-test failure votes 1-4 Bit 15=Voter Fail Latch Bit 16=Timeout Latch Bit 17=Voter Termination Control Latch Bit 18=Timer Termination Control Latch Bit 19=IOP Transmission Termination Bits 20-31=Watchdog Timer value.	Failure of the IOP, the R/M logic or the R/M Status Register.
UNEXPECTED IOP PROGRAMMED INTERRUPT **ERROR CODE: 141**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: G2=0808 0000 G3=INTC.	An unexpected External 2 (EX2) Interrupt has been generated and the contents of Interrupt Register C (INTC) indicates that either an invalid bit is set, more than one valid bits are set or no bits are set at all. The EX2 (MSC Program) interrupts are a priority of interrupts generated from within an MSC program by executing a programmable Interrupt (@INT) instruction. Upon detection of the error condition, the MSC program will set specific bits in INTC to '1' and execute the @INT instruction to interrupt the CPU. The CPU will then generate the EX2 interrupt, store the current Program Status Word (PSW) into low core locations 88-8B and execute the EX2 Interrupt handler. The self-test program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register C format: (0-no Interrupt, 1-interrupt) Bits 0-11=MSC Program Interrupts 1-12, respectively Bits 12-31=Zeroes (0).	Failure of the IOP, the MSC or INTC.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
IOP ROS PARITY TEST FAILED **ERROR CODE: 1 4 2**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R2=MSC start address 2 and 3. R3=interrupt code/ Interrupt Reg A R6=MSC program counter.	A test of the IOP Read Only Storage (ROS) parity interrupt logic has failed. The test initiates an MSC program which executes the '@STP 2' instruction, which attempts to perform a microinstruction that contains bad (even) parity. This should cause the IOP to generate an External 0 (EX0) interrupt, the MSC to enter a halt state, bit 3 of Interrupt Register A (INTA) to be set and the current Program Status Word (PSW) to be stored into low core locations 78-7B. The error can be annunciated for any of the following 3 reasons: 1. the EX0 interrupt never occurred, 2. bit 3 of INTA was not set to '1' or the interrupt code in the PSW was not '0000', or 3. the MSC did not go into a halt state following detection of the parity error. The program responses for 1-3 above are as follows: 1. annunciate the error, reset the interrupt return address, set bit 15 of General Register 5 (R5) to '1' and continue, 2. annunciate the error, set bit 12 of R5 to '1' and continue, 3. annunciate the error, halt the MSC, set bit 14 of R5 to '1' and continue.	Failure of the CPU or IOP, the IOP Control/Monitor logic, INTA or the MSC.
THE INSTRUCTION MONITOR IS NOT MASKABLE **ERROR CODE: 1 4 3**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The CPU generated an Instruction Monitor (IM) interrupt while the proper mask was set in the current Program Status Word (PSW). An IM interrupt is generated by the CPU when an attempt is made to execute an unprotected memory location as program code. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'DF0C' and setting the store protect bit of a branch instruction to '0' (unprotected). The value 'DF0C' replaces bits 32-47 of the current PSW, thus setting the IM mask (bit 34) to '0'. This will enable the generation of all interrupts except the IM. If the IM interrupt occurs following execution of the unprotected branch, the current PSW is stored at low core locations 70-73. The self-test program will annunciate the error, set the branch instruction store protect bit to '1' (protected) and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
THE INS MON DID NOT INTRPT ON A UNPROT INSTRCTN **ERROR CODE: 144**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=IM interrupt return address.	The CPU did not generate an Instruction Monitor (IM) interrupt following an attempt by the program to execute an unprotected branch instruction. An IM interrupt is generated by the CPU when an attempt is made to execute an unprotected memory location as program code. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'FF0C' and setting the store protect bit of a branch instruction to '0' (unprotected). The value 'FF0C' replaces bits 32-47 of the current PSW, thus setting the IM mask (bit 34) to '1'. This will enable the generation of all interrupts, including the IM. If the IM interrupt does not occur following execution of the unprotected branch, the self-test program will annunciate the error, reset the interrupt return address and continue.	Failure of the CPU.
IOP RET NOT 0 AFTER EXT 1 PCI DATA PARITY INT **ERROR CODE: 145**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=INTB.	The IOP has incorrectly set Interrupt Register B (INTB) to a non-zero value following an External 1 (EX1) Program Controlled Input (PCI) data parity interrupt. INTB is utilized to identify the External 1 (EX1) interrupts which are generated by the IOP. The PCI data parity interrupt is generated by the CPU following the detection of bad (even) parity in the data transferred from the IOP during a PCI operation. The self-test program verifies the interrupt logic by loading the value of INTB, returned from the EX1 interrupt handler routine, into General Purpose Register 4 (R4). A check is then made to determine if the returned value is equal to zero. If not, the self-test program will annunciate the error and continue. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP, the IOP Control/Monitor logic or INTB.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
NO EXPECTED EXT 1 INT AFTER PCI DATA PARITY **ERROR CODE: 146**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=0804 0000.	The CPU did not generate an expected External 1 (EX1) interrupt following a test of the CPU parity checker circuitry. The test issues a Program Controlled Output (PCO) instruction to the IOP to command it to generate bad (even) parity. The CPU will then issue a Program Controlled Input (PCI) instruction to the IOP to read the contents of Interrupt Register B (INTB). Since the IOP was commanded to generate even parity, the contents of INTB will contain bad parity when transferred to the CPU. Upon detection of the parity error, the CPU should generate an EX1 PCI Data Parity Interrupt, with interrupt code '0001' and the current Program Status Word (PSW) stored into low core locations 80-83. If no interrupt occurs, the self-test program will annunciate the error and continue.	Failure of the CPU Interrupt logic, the CPU parity checker circuitry or the IOP PCI parity generator.
INT CODE NOT 2 AFTER EXT 1 PCI DATA PARITY INT **ERROR CODE: 147**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=0002 0000.	The CPU has generated an External 1 (EX1) Direct Memory Access (DMA) data parity interrupt with an incorrect interrupt code in the Program Status Word (PSW). An interrupt code of '0002' is generated by the CPU following the detection of even parity in data transferred from the IOP during a DMA 'In' operation (main memory write), and is stored along with the current PSW into low core locations 80-83. The self-test program verifies the interrupt logic by loading the known value of the interrupt code into General Purpose Register 4 (R4). A check is then made to determine if the known value is equal to the contents of location '83'. If not, the self-test program will annunciate the error and continue.	Failure of the CPU interrupt logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MEMORY PARITY TEST - PARITY ERROR **ERROR CODE: 148**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R2=start address of 16 halfword read R3=interrupt return address G4=interrupt return address; 2. R2=start address of 4 halfword read. G4=interrupt return address.	An unexpected Machine Check (MC) Interrupt has been generated by the CPU following detection of a bad parity during a test of GPC main memory. The test, which performs a sequential read (case 1) of every GPC memory location and selected reads (case 2), has encountered even (bad) parity in either the address specified for a memory access, or the data returned from CPU/IOP memory. The error can be detected by either the CPU or IOP, but the MC interrupt is always generated by the CPU. The appropriate interrupt code will be generated and stored along with the current Program Status Word (PSW) into low core locations 40-43. The self-test program will annunciate the error and continue.	Failure of the CPU or main memory.
DEU BITE ERROR **ERROR CODE: 149**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=DEU response Headerword, R5=DEU SWSW, R6=DEU BITE Status Word 1 and 2.	The Display Electronics Unit (DEU) has detected a critical BITE error. The self-test program, during the DEU Poll Response routine, has found bit 14 of the DEU poll response headerword to be set to '1' (critical BITE error present). The routine will then test for the following conditions: 1. if a power transient occurred or the Standalone Self-Test Program is in progress, it is assumed to have been induced and control is passed back to the job scheduler; 2. if a power transient did not occur, a check is made for a CPU software fail or a keyboard channel A or B fail (any of bits 7-9 in the DEU BITE Status register 2 equal '1'); 3. if the BITE error has not been indicated on 2 consecutive poll responses from the DEU, control is passed back to the job scheduler. Otherwise, the self-test program will save the contents of DEU BITE status registers 1 and 2 into General Register 6, annunciate the error and return to the job scheduler.	Failure of the DEU

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
STARTUP SELF - TEST ABENDED WITH TOO MANY ERRORS **ERROR CODE: 150**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The Self-Test Program has annunciated too many errors during the initial program execution, and the self-test of the GPC has been aborted. The maximum number of errors allowable for the first execution of the STP is 98. Subsequent executions of the program are not subject to this limit, and an overwrite of errors entered earlier in the stack is initiated after the 100th error is encountered. The self-test program will terminate self-testing, annunciate the error and set up for realtime execution of the self-test program.	Failure of the CPU, the IOP or main memory.
MMU COMPUTED CHECK SUM MISCOMPARE **ERROR CODE: 151**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The CPU has detected a miscompare between the checksum supplied by the Mass Memory Unit (MMU) and the one it has generated, during the transfer of a block of data. The CPU adds all halfword data locations in the block, excluding parity and storage protection bits, transferred from the MMU during a PASS, BFS, Backup System Loader (BSL) or DEU software load. The checksum is one method of validating the contents of the block.	Failure of the MMU or BCE, or an incorrect checksum supplied by the MMU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU WILL NOT GO READY **ERROR CODE: 152**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=08CA 0000 R5=DIA register.	The CPU has determined that the Mass Memory Unit (MMU) 1 and 2 ready discretes in the Discrete Input A (DIA) register do not indicate that either MMU is available for use. The discretes are statused prior to a PASS, BFS, Backup System Loader (BSL) or DEU software load. The CPU issues a Program Controlled Input (PCI) instruction to read DIA register in the IOP. A check is then made to verify that the ready discrete of the selected MMU (bit 6-MMU1, bit 7-MMU2) is set to '1'. The CPU allows 90 seconds for the MMU to set the discrete following its selection via the IPL Source Select Switch. If the time expires before the discrete is set, the self-test program will deschedule the job, annunciate the error and return to the job scheduler. DIA Register format: Bit 0=Halt Bit 1=Standby Bit 2=Run Bit 3=IPL Bit 4=MMU 1 IPL Bit 5=MMU 2 IPL Bit 6=MMU 1 Ready Bit 7=MMU 2 Ready Bit 12=Inhibit Channels 10-13 Bit 13=Inhibit Channels 14-17 and 20-23	Failure of the MMU or the Control/Monitor logic.
MMU EXPECTED POSITION ERROR **ERROR CODE: 153**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=actual MMU position R6=MMU position computed by the CPU.	The CPU has determined that either Mass Memory Unit 1 or 2 is positioned incorrectly during a PASS, BFS, Backup System Loader (BSL) or DEU software load. The CPU checks the MMU's current File-Track-Subfile position against the position it has computed following each position and read transaction. A check is then made to verify that the values agree. If not, the self-test program will reset the MMU interface variables, annunciate the error and return to the job scheduler.	Failure of the MMU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU SOURCE SELECT SWITCH POSITION ERROR **ERROR CODE: 154**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R5=DIA register.	The CPU has determined that the Mass Memory Unit (MMU) 1 and 2 IPL source select discretes in the Discrete Input A (DIA) register do not indicate that an MMU BCE is selected for the software load. The discretes are statused prior to a PASS, BFS, Backup System Loader (BSL) or DEU software load. The CPU issues a Program Controlled Input (PCI) instruction to read DIA register in the IOP. A check is then made to verify that one of the IPL source select discretes is set to '1' (bit 4-MMU1, bit 5-MMU2). The self-test program will deschedule the job, reset the MMU interface variables, annunciate the error and return to the job scheduler. DIA Register format: Bit 0=Halt Bit 1=Standby Bit 2=Run Bit 3=IPL Bit 4=MMU 1 IPL Bit 5=MMU 2 IPL Bit 6=MMU 1 Ready Bit 7=MMU 2 Ready Bit 12=Inhibit Channels 10-13 Bit 13=Inhibit Channels 14-17 and 20-23.	Failure of the IPL Source Select Switch, the Control/Monitor logic, the DIA register or an incorrect setting of the IPL Source Select Switch.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU BITE STATUS REG A OR B ERROR **ERROR CODE: 155**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=MMU BITE Status Registers A and B.	The CPU has detected a Mass Memory Unit (MMU) 1 or 2 BITE error during an MMU I/O transaction. The CPU reads MMU Status Registers A and B and checks for any bits to be set. If so, the self-test program will reset the MMU interface variables, annunciate the error and return to the job scheduler on the second occurrence of the BITE error. MMU Status Register A format: (0 - no failure, 1 - failure) Bit 0=Power transient Bit 1=File address not equal Bit 2=Command error Bit 3=Write protect violation Bit 4=Invalid Op code MMU Status Register B format: (0 - no failure, 1 - failure) Bit 0=Data Count word low Bit 1=Read tape data dropout Bit 2=read chk assurance lost Bit 3=Read tape Parity Error Bit 4=MIA invalid Manchester Code Bit 5=Command Rec'd/not ready Bit 6=EOF Block Count zero Bit 7=No search address compare Bit 8=Malfunction Bit 9=BOT sensed Bit 10=EOF sensed Bit 11=power supply out of tolerance - operation aborted Bit 12=MIA bit count error Bit 13=MIA parity error Bit 14=Invalid 101 check Bit 15=MIA data/address error	Failure of the MMU, the MIA or incorrect inputs to the MMU transaction.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU BCE I/O ERROR **ERROR CODE: 156**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R2=Go/No-Go register R3=1000 0000; R4=BCE number 2. R3=1000 0000 R4=Go/No-Go register.	The CPU has encountered a BCE I/O error during either of 2 cases: 1. during the Mass Memory Unit (MMU) scheduler routine to verify the MMU's state prior to initiating an I/O transaction, or 2. during a routine which performs a checksum of each block of data transferred from the MMU to the GPC. The program reads the Go/No-Go Register using a Program Controlled Input (PCI) instruction, then checks bits 18 and 19 (MMU BCEs) for a value of '1'. If any equal '0', indicating that the necessary BCE is in a No-Go state, the self-test program will annunciate the error and continue. Go/No-Go Register format: (0-NoGo, 1-Go) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of a BCE or the MIA.
MSC STATUS ERROR AFTER BAD ADDRESS PARITY TEST **ERROR CODE: 157**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R5=220B 8000 R7=MSC Status register. NOTE: The MSC Status register is a 17-bit register. Bits 14- 31 of R7 correspond to this register.	The IOP failed to correctly set the appropriate bits in the MSC Status Register during a test of the CPU parity checker circuitry. The test forces an address parity error to occur during an instruction fetch, which should cause bit 15 (illegal op code) and bit 16 (Go/No-Go) to be set to '1' in the register. A subsequent read of the MSC Status register by the self-test program, using a Program Controlled Input (PCI) instruction, has found that bit 15 and 16 remain equal to zero following the generation of the parity error. The self-test program will annunciate the error and continue. MSC Status Register format: Bits 0-7=Undefined Bit 8=self-test failure Bits 9-10=Undefined Bit 11=SIO (Start BCE execution) error Bit 12=LBP (load BCE program counter) error Bit 13=LBB (load BCE base register) error Bit 14=boundary alignment error Bit 15=illegal operation code Bit 16=Status 1 register bit (0-Go, 1-No-Go) Bit 17=Busy/Wait register bit (0-wait, 1-busy)	Failure of the MSC, the IOP Control/Monitor logic or the MSC Status Register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MSC STAT 1 ERROR AFTER BAD ADDRESS PARITY TEST **ERROR CODE: 158**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=FFFF FF80 R5=Go/No-Go register R6=1000 0000 R7=8000 0000.	A Program Controlled Output (PCO) instruction has failed to reset MSC bit of the Status 1 (Go/No-Go) Register. The bit was set to '0' (No-Go) when an address parity error was forced during an instruction fetch, while testing the CPU parity checker circuitry. The parity error causes an illegal op code to be encountered by the MSC, which in turn will set its Go/No-Go bit to '0', indicating that it has encountered an error. A subsequent read of the Go/No-Go register by the self-test program, using a Program Controlled Input (PCI) instruction, has found that the MSC bit remains equal to zero following execution of the PCO instruction intended to reset it. The self-test program will annunciate the error and continue. Go/No-Go Register format: (0-NoGo, 1-Go) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the MSC, the Go/No-Go Register or the IOP Control/Monitor logic.
INSTRUCTION MONITOR INTERRUPT ON PROTECTED INST. **ERROR CODE: 159**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The CPU erroneously generated an Instruction Monitor (IM) interrupt following an attempt by the program to execute a protected branch instruction. An IM interrupt should only be generated by the CPU when an attempt is made to execute an unprotected memory location as program code. The self-test program verifies the interrupt logic by executing a Set System Mask (SSM) instruction using the value of 'FF0C' and setting the store protect bit of a branch instruction to '1' (protected). The value 'FF0C' replaces bits 32-47 of the current PSW, thus setting the IM mask (bit 34) to '1'. This will enable the generation of all interrupts, including the IM. If the IM interrupt occurs following execution of the protected branch, the current PSW is stored at low core locations 70-73. The self-test program will then annunciate the error, reset the interrupt return address and continue.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 1 C A16 **ERROR CODE: 160**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 1 of main memory (sector 0, locations '0000-7FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 1.
POWER ON SWITCHING TEST FAILURE PAGE 2 C A15 **ERROR CODE: 161**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 2 of main memory (sector 0, locations '0000-7FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 2.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 3 C A18 **ERROR CODE: 162**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 3 of main memory (sector 1, locations '8000-FFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 3.
POWER ON SWITCHING TEST FAILURE PAGE 4 C A17 **ERROR CODE: 163**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 4 of main memory (sector 1, locations '8000-FFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 4.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 5 C A20 **ERROR CODE: 164**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 5 of main memory (sector 2, locations '10000-17FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 5.
POWER ON SWITCHING TEST FAILURE PAGE 6 C A19 **ERROR CODE: 165**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 6 of main memory (sector 2, locations '10000-17FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 6.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 7 C A22 **ERROR CODE: 166**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 7 of main memory (sector 3, locations '18000-1FFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 7.
POWER ON SWITCHING TEST FAILURE PAGE 8 C A21 **ERROR CODE: 167**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 8 of main memory (sector 3, locations '18000-1FFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 8.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 9 C A24 **ERROR CODE: 168**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 9 of main memory (sector 4, locations '20000-27FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 9.
POWER ON SWITCHING TEST FAILURE PAGE 10 C A23 **ERROR CODE: 169**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 10 of main memory (sector 4, locations '20000-27FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 10.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 11 I A27 **ERROR CODE: 170**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 11 of main memory (sector 5 lower, locations '28000-2BFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 1.
POWER ON SWITCHING TEST FAILURE PAGE 12 I A26 **ERROR CODE: 171**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 12 of main memory (sector 5 lower, locations '28000-2BFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 2.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 13 I A25 **ERROR CODE: 172**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 13 of main memory (sector 5 upper, locations '2C000-2FFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 3.
POWER ON SWITCHING TEST FAILURE PAGE 14 I A24 **ERROR CODE: 173**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 14 of main memory (sector 5 upper, locations '2C000-2FFFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 4.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER ON SWITCHING TEST FAILURE PAGE 15 I A23 **ERROR CODE: 174**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 15 of main memory (sector 6 lower, locations '30000-33FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 5.
POWER ON SWITCHING TEST FAILURE PAGE 16 I A22 **ERROR CODE: 175**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 16 of main memory (sector 6 lower, locations '30000-33FFF') has failed the Power On Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power On test makes 50 accesses of the memory page, one every 12.9 microseconds, for an expected cycle time of 645 microseconds. If the power supply to the memory page is placed into the Quiescent state at any time during the test, the expected cycle time will be exceeded and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 6.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 1 C A16 **ERROR CODE: 176**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 1 of main memory (sector 0, locations '0000-7FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 1.
POWER OFF SWITCHING TEST FAILURE PAGE 2 C A15 **ERROR CODE: 177**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 2 of main memory (sector 0, locations '0000-7FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 2.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 3 C A18 **ERROR CODE: 178**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 3 of main memory (sector 1, locations '8000-FFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 3.
POWER OFF SWITCHING TEST FAILURE PAGE 4 C A17 **ERROR CODE: 179**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 4 of main memory (sector 1, locations '8000-FFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 4.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 5 C A20 **ERROR CODE: 180**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 5 of main memory (sector 2, locations '10000-17FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 5.
POWER OFF SWITCHING TEST FAILURE PAGE 6 C A19 **ERROR CODE: 181**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 6 of main memory (sector 2, locations '10000-17FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 6.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 7 C A22 **ERROR CODE: 182**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 7 of main memory (sector 3, locations '18000-1FFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 7.
POWER OFF SWITCHING TEST FAILURE PAGE 8 C A21 **ERROR CODE: 183**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 8 of main memory (sector 3, locations '18000-1FFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 8.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 9 C A24 **ERROR CODE: 184**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 9 of main memory (sector 4, locations '20000-27FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 9.
POWER OFF SWITCHING TEST FAILURE PAGE 10 C A23 **ERROR CODE: 185**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 10 of main memory (sector 4, locations '20000-27FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.4 microseconds, for an expected cycle time of 1295 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of CPU memory page 10.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 11 I A27 **ERROR CODE: 186**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 11 of main memory (sector 5 lower, locations '28000-2BFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.5 microseconds, for an expected cycle time of 1300 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 1.
POWER OFF SWITCHING TEST FAILURE PAGE 12 I A26 **ERROR CODE: 187**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 12 of main memory (sector 5 lower, locations '28000-2BFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.5 microseconds, for an expected cycle time of 1300 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 2.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 13 I A25 **ERROR CODE: 188**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 13 of main memory (sector 5 upper, locations '2C000-2FFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.5 microseconds, for an expected cycle time of 1300 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 3.
POWER OFF SWITCHING TEST FAILURE PAGE 14 I A24 **ERROR CODE: 189**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 14 of main memory (sector 5 upper, locations '2C000-2FFFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.5 microseconds, for an expected cycle time of 1300 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 4.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
POWER OFF SWITCHING TEST FAILURE PAGE 15 I A23 **ERROR CODE: 190**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 15 of main memory (sector 6 lower, locations '30000-33FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.5 microseconds, for an expected cycle time of 1300 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 5.
POWER OFF SWITCHING TEST FAILURE PAGE 16 I A22 **ERROR CODE: 191**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=memory page number.	Page 16 of main memory (sector 6 lower, locations '30000-33FFF') has failed the Power Off Switching Test. To preserve power consumption, each memory page pair contains a controller which will switch power between On and Off (Standby and Quiescent), depending on the time between accesses. The switching limit for the AP-101B hardware is 19 +/- 6 microseconds. Therefore, if the memory page is accessed every 13 microseconds or less, power should remain in Standby. If the page is not accessed for more than 25 microseconds, the page is placed into a low-power Quiescent state. The Power Off test makes 50 accesses of the memory page, one every 25.5 microseconds, for an expected cycle time of 1300 microseconds (includes 1/2 microsecond per access for the expected switch from Standby to Quiescent). If the power supply to the memory page remains in Standby between any 2 accesses, the cycle time will be less than expected and the test will be considered failed. The self-test program will annunciate the error and continue with the testing of the next memory page.	Failure of the One-shot Controller in the Power Switch Control logic of IOP memory page 6.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
BCE WRAP-#STP ERR -AT LEAST 1 BCE DID NOT FINISH **ERROR CODE: 192**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=FFFF FF80 R6=1004 0000 R7=logical 'AND' of R2 and the Busy/Wait register.	At least one of the tested BCE processors is still busy following completion of the IOP Processor Self-Test routine, #STP. The BCE's addressing logic, data flow, Direct Memory Access (DMA) logic, channel queues and MIA circuitry are tested using the #STP instruction. After a pre- determined amount of time, each BCE processor is checked to verify that it is in a wait state, which indicates that it has successfully completed its respective self-test. The self-test program checks the results of the routine by reading the Status 4 (Busy/Wait) register, using a Program Controlled Input (PCI). The register is then 'AND'd with the bit string 'FFFF FF80'. If the result is not zero, the self-test program will annunciate the error and continue Busy/Wait Register format: (0-wait, 1-busy) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of the IOP or a BCE.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MSC-@STP WRAP ERROR- MSC NOT IN WAIT **ERROR CODE: 193**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0000 0001.	The MSC was not indicated to be busy during execution of the IOP Processor self-test routine, @STP. The @STP routine tests the MSC's addressing logic, data flow, arithmetic logic, parity detection circuitry, channel operation and failure vote logic. A portion of the arithmetic logic is tested by executing a Load MSC Status (@LMS) instruction and a store instruction, which stores the current value of the MSC Status register into a program variable. A check is then made to verify that bit 17 of the register is equal to '1', indicating that the MSC is busy executing a program. If the bit equals '0', the self-test program will annunciate the error and continue. MSC Status Register format: Bits 0-7=Undefined Bit 8=self-test failure Bits 9-10=Undefined Bit 11=SIO (Start BCE execution) error Bit 12=LBP (load BCE program counter) error Bit 13=LBB (load BCE base register) error Bit 14=boundary alignment error Bit 15=illegal operation code Bit 16=Status 1 register bit (0-Go, 1-No-Go) Bit 17=Busy/Wait register bit (0-wait, 1-busy)	Failure of the IOP, the MSC or the MSC Status register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
IOP INTERRUPT REG A OR B HAVE STUCK BITS **ERROR CODE: 194**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=INTA R4=logical 'OR' of INTA and INTB R6=0804 0000.	The IOP was not able to correctly reset either Interrupt Register A or B (INTA or INTB). The IOP logic is designed to reset the interrupt registers to zero after their contents are read by the CPU. The self-test program reads each register using Program Controlled Input (PCI) instructions, masks off all insignificant bits and places the logical 'OR' of the contents of the two registers into General Register 4 (R4). If R4 will contains a non-zero value, either INTA or INTB failed to be reset. In this case, the self-test program will annunciate the error and continue. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Not used. Held at a zero (0) state Bit 4=DMA Queue Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP, INTA, INTB or the Control/ Monitor logic.
IOP INTERRUPT REG C HAS STUCK BITS **ERROR CODE: 195**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=INTC R6=0808 0000.	The IOP was not able to correctly reset Interrupt Register C (INTC). The IOP logic is designed to reset the interrupt register to zero after its contents are read by the CPU. The self-test program reads INTC using a Program Controlled Input (PCI) instruction, masks off bits 12-31 and places contents of the register into General Register 3 (R3). If R3 contains a non-zero value, INTC has failed to be reset. In this case, the self-test program will annunciate the error and continue. Interrupt Register C format: (0-no interrupt, 1-interrupt) Bits 0-11=MSC Program Interrupts 1-12, respectively Bits 12-31=Zeroes (0).	Failure of the IOP, INTC or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
IOP PROCESSORS ENABLE TEST FAILED **ERROR CODE: 196**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=enable control word R5=Processor Halt register R6=040C 0000.	A Program Controlled Output (PCO) instruction has failed to correctly initialize all IOP processors, during a test of the IOP Control/Monitor (C/M) logic. The self-test program issues a PCO instruction with a control word of 'FFFF FF80' to disable all IOP processors, then issues a second PCO instruction using a control word which will enable only the processors which were not previously disabled. A read of the Status 5 (Processor Halt) register following the test, using a Program Controlled Input (PCI) instruction, has found that the MSC or at least one of the BCE processors remains disabled following execution of the PCO instruction intended to enable it. The self-test program will annunciate the error and continue. Processor Halt Register format: (0-disabled, 1-enabled) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined. Control Word format: (0-ignore, 1-enable/disable) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor. Bits 26-31=Undefined.	Failure of the IOP, the Control/Monitor logic or the Processor Halt register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
BCE#25 FAILED @STP WRAP TEST **ERROR CODE: 197**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R3=9204 0000 R4=Processor Halt reg R5=MSC Program Counter; 2. R4=0004 7FFF R5=A0+B1+C1 R6=2390 0000	The CPU has determined that either of two cases exists: 1) a Program Controlled Output (PCO) instruction has failed to halt all IOP processors, or 2) a test of the local storage addressing for BCE #25 has failed. In case 1) above, the self-test program has read the contents of the Status 5 (Processor Halt) register and determined that at least one IOP processor remains enabled, following execution of the PCO intended to halt all processors . In case 2) above, the self-test program has attempted to verify the results of the BCE self-test routine (#STP) by computing the sum of locations A0, B1 and C1 in the local storage area of BCE #25. The value should equal '0004 7FFF'. The self-test program will annunciate the error and continue. Processor Halt Register format: (0-disabled, 1-enabled) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of BCE #25 local store logic or the Processor Halt register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
BCE TRANS/RECV REGS LOAD/READ/DISABL TEST FAILED **ERROR CODE: 198**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=Trans/Rec bit string R5=0404 0000 R6=0400 0000.	The CPU is unable to correctly control the MIA Transmitters and Receivers using Program Controlled Input and Output (PCI/O) Instructions. First, the transmitter and receiver status registers (TX and RX) are stored. Then, the self-test program checks the capability of the CPU to control the transmitters and receivers by issuing PCI and PCO instructions in the following sequence: enable MIA transmitters; enable MIA receivers; read the TX register; read the RX register; disable MIA transmitters; disable MIA receivers. General Register 2 (R2) is utilized in such a way that a failure to correctly configure a transmitter or receiver in any portion of the sequence results in a zero being stored into the appropriate bit position in R2. The self-test program will annunciate the error, restore the contents of the TX and RX registers to their pre-test configurations and continue. General Register 2 format: (0-failure, 1-no failure) Bit 0=MSC Bits 1-24=MIA Transmitter/Receiver 1-24, respectively Bits 25-31=Undefined.	Failure of the IOP or CPU, the Control/Monitor logic or a MIA transmitter/ receiver.
IOP SELF TEST MSC PROGRAM COUNTER INCORRECT **ERROR CODE: 199**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=MSC Program Counter R3=2201 0000.	The MSC is in a wait state at an incorrect memory location. This occurs during execution of the MSC/BCE self-test routines @STP and #STP, which verify the proper function of the processors using built-in test instructions. The self-test routines will place the MSC or a BCE into a wait state upon detection of an error, via the @WAT command. This error indicates that the wait state was invoked, but not by using the @WAT command. Following completion of the CPU test, the program reads the MSC Program Counter, using a Program Controlled Input (PCI) instruction, to verify that the MSC is in a wait state at the correct memory location. If not, the program annunciates the error, saves general purpose registers 0-7 and continues.	Failure of the IOP, MSC, a BCE or the @STP or #STP logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
ILLEGAL IOP TERMINATE A DIA BIT 12=1 **ERROR CODE: 200**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=0818 0000 R3=DIA register shifted right 2 places.	The CPU has determined that the IOP Terminate A discrete is set to an incorrect value. The CPU issues a Program Controlled Input (PCI) instruction to the IOP to read the Discrete Input A (DIA) register. This register contains both the I/O Terminate A and B discretes, which are used to determine which BCEs the computer is able to transmit commands on, according to the following table: If A=0, B=0 - BCEs 1-24 are tested If A=0, B=1 - BCEs 1-13,18,19 and 24 are tested If A=1, B=0 - BCEs 1-9 and 14-24 are tested If A=1, B=1 - BCEs 1-9,18,19 and 24 are tested. During the self-test program, the values of the 2 discretes should both be '0' so that all BCEs take part in the testing. If I/O Term A is found to equal '1', the self-test program will set a flag to allow only one annunciation of the error, annunciate the error and continue.	Failure of the IOP or the Control/Monitor logic.
>>>GPC POWER REFAIL-PROGRAM/MACHINE WERE RESET **ERROR CODE: 201**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The GPC has successfully recovered from a power failure which required the computer to be reset. Upon detection of a power transient (i.e. voltage drops below 17.5), the CPU will perform a Power Down Putaway routine, which utilizes a 100 microsecond window to store each of the general register sets, the clock values and the contents of other program variables necessary to restore the computer to the configuration it was in prior to the failure. The next time the computer is powered on, or if the power failure was transient, control is passed to the Power Fail Executive routine using a branch through the Power On Program Status Word (PSW, low core location '0004'). The routine, made up of 12 separate modules, will attempt to determine where in the program the power failure occurred and what steps need to be taken to recover. In this case, the routine could not determine what needed to be done to recover, so the self-test program will store the SVC number of this error (201) for later annunciation, re-perform the one shot self-test program and re-initialized the computer for real-time execution.	Failure of the GPC 28V power supply.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
>>>GPC POWER REFAIL -FULL RECOVERY SUCCESSFUL **ERROR CODE: 202**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The GPC has successfully recovered from a second power failure which occurred during the processing of the initial power failure recovery steps. Upon detection of a power transient (i.e. voltage drops below 17.5), the CPU will perform a Power Down Putaway routine, which utilizes a 100 microsecond window to store each of the general register sets, the clock values and the contents of other program variables necessary to restore the computer to the configuration it was in prior to the failure. The next time the computer is powered on, or if the power failure was transient, control is passed to the Power Fail Executive routine using a branch through the Power On Program Status Word (PSW, low core location '0004'). The routine, made up of 12 separate modules, will determine where in the program the power failure occurred and what steps need to be taken to recover. In this case, the second power failure occurred while the Power Fail Executive was executing. The self-test program will only store the SVC number of this error (202) for later annunciation and continue with the recovery procedure.	Failure of the GPC 28V power supply.
GPCIPL RR.VV.PP.II.MM LOADED **ERROR CODE: 203**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates which version of the GPC IPL program is loaded, where: RR = Revision number, VV = Version number, PP = Patchset number, I I = I-load set number, MM = mass memory area number. The self-test program will annunciate the message, set a flag to allow 1 annunciation and continue.	Not an error condition.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
GPC ID CHANGED FROM ## TO %% ERROR CODE: 204	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The GPC Identifier number contained in the Discrete Input B (DIB) register has changed since the last time a Display Electronics Unit (DEU) was polled. The ID is determined by a hardwire connector and is set according to the physical position of the GPC. Each time the DEU poll routine is executed, the CPU issues a Program Controlled Input (PCI) instruction to the IOP to read the DIB register. Bits 0-2 of the register are then checked to verify that they are equal to the ID stored when the computer was first powered on or when it last updated. The self-test program will convert the new and old IDs into a format which can be displayed on the CRT, insert these values into the message, annunciate the error, save the new ID for subsequent checks and continue. NOTE: The dynamic parts of this message, ## and %%, will contain the old and new GPC ID's, respectively.	Failure of the DIB register, the Control/Monitor logic or the GPC hardwire connector.
AT LEAST 1 TESTED BCE DID NOT SET ITS IND_BIT ERROR CODE: 205	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=MSC Program Counter R3=2201 0000.	The MSC timed out while waiting for all tested BCEs to set their respective indicator bits in the MSC-BCE Indicator Register. This occurs during execution of the MSC/BCE self-test routines @STP and #STP, which verify the proper function of the processors using built-in test instructions. Part of the BCE self-test routine is to perform an instruction which sets its respective indicator bit to '1'. The indicator has no hardware defined meaning, and can be used for any programmer-specified convention. The MSC allows 33 microseconds for all BCEs to set their respective bit. If this timer expires, the MSC enters a wait state at an unexpected memory location. Following completion of the CPU test, the program reads the MSC Program Counter, using a Program Controlled Input (PCI) instruction, to verify that the MSC is in a wait state at the correct memory location. If not, the program annunciates the error, saves general purpose registers 0-7 and continues.	Failure of a BCE or #STP logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MSC DETECTED AT LEAST 1 TESTED BCE IS STILL BUSY **ERROR CODE: 206**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=MSC Program Counter R3=2201 0000.	The MSC has timed out while waiting for all tested BCEs to return to the Wait State and is itself in a Wait State at an unexpected memory location. This occurs during execution of the MSC/BCE self-test routines @STP and #STP, which verify the proper function of the processors using built-in test instructions. Part of the BCE self-test routine calls for the BCE to be placed into the wait state after completion of the #STP instruction. The MSC allows 33 microseconds for all BCEs to set their respective Busy/Wait Register bit to '0' (wait). If this timer expires, the MSC enters a wait state at an unexpected memory location. Following completion of the CPU tests, the program reads the MSC Program Counter, using a Program Controlled Input (PCI) instruction, to verify that the MSC is in the wait state at the correct memory location. If not, the program annunciates the error, saves general purpose registers 0-7 and continues.	Failure of a BCE or #STP logic.
AT LEAST 1 TESTED BCE COULD NOT RESET ITS INDCTR **ERROR CODE: 207**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=MSC Program Counter R3=2201 0000.	The MSC timed out while waiting for all tested BCEs to reset their respective indicator bits in the MSC-BCE Indicator Register. This occurs during execution of the MSC/BCE self-test routines @STP and #STP, which verify the proper function of the processors using built-in test instructions. Part of the BCE self-test routine is to perform an instruction which resets its respective indicator bit to '0'. The indicator has no hardware defined meaning, and can be used for any programmer-specified convention. The MSC allows 33 microseconds for all BCEs to reset their respective bit. If this timer expires, the MSC enters a wait state at an unexpected memory location. Following completion of the CPU test, the program reads the MSC Program Counter, using a Program Controlled Input (PCI) instruction, to verify that the MSC is in a wait state at the correct memory location. If not, the program annunciates the error, saves general purpose registers 0-7 and continues.	Failure of a BCE or #STP logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
DCP RR.VV.PP.II.MM LOADED **ERROR CODE: 208**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=starting address of the DCP in GPC memory.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates which version of the Display Control Program (DCP) is loaded, where: RR = Revision number, VV = Version number, PP = Patchset number, I I = I-load set number, MM = mass memory area number. A copy of the Display Control Program (DCP) is loaded into GPC memory during the Initial Program Load (IPL) process. The 12th halfword of the DCP contains the current version/revision of the program. The self-test program reads this location, converts the data into a format which can be displayed on the CRT and places the information into the advisory message. The self-test program will annunciate the message and continue.	Not an error condition.
ILLEGAL GPC/CRT ID ENTRY **ERROR CODE: 209**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: Case 1. R2=DEU specified R4=GPC specified; 2. R2=DEU specified R4=GPC specified; 3. R4=DEU specified R6=GPC specified.	A 'GPC/CRT XY EXEC' command has failed for one of the following 3 reasons: 1. the Display Electronics Unit (DEU) which was to be deassigned (Y) was not being controlled by the GPC which is executing the self-test program, 2. the GPC number specified in the command (X) is not equal to the value of the GPC ID in the Discrete Input B (DIB) register, or 3. the GPC or DEU number specified in the command (X or Y) is out of valid range (i.e X not equal to 0-5, Y not equal to 1-4). The check for the 'GPC/CRT' keyboard entry is made during the DEU poll routine. If any of the above conditions exist, the self-test program will annunciate the error, ignore the command and continue.	Failure to correctly specifiy a GPC or CRT number during a CRT de-assign/re-assign.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
SELECTED FORMAT WAS NOT IN INITIAL LOAD **ERROR CODE: 210**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=GPC address of table R3=DEU format number (0-10) R4=GPC address of DEU Critical Formats ('FFFF 0000' if invalid)	The CPU has determined that an 'ITEM 17+X EXEC' command entered on the GPCIPL Menu Display is invalid due to an incorrect format number specification (X). The value is used as an index into a table of ten main memory locations, two of which contain copies of the Standalone Self-Test Program (SASTP) and the DEU Critical Formats. Although the valid range of the format number is zero through hex 'A', only one critical format load (X=1) and the SASTP (X=A) are currently available. The self-test program will annunciate the error and continue.	Incorrect format number was specified for a Critical Format Load keyboard item entry.
DESIRED SYSTEM SOFTWARE NOT ALLOCATED ON THE MMU **ERROR CODE: 211**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The CPU has determined that an unavailable PASS or BFS software load has been requested from the GPCIPL Menu Display. Although the GPCIPL Menu Display provides for up to 8 copies each of PASS and BFS, there are only 3 copies resident on Mass Memory. Therefore, only items 1,3 and 5 are valid for a PASS load, and 2,4 and 6 for a BFS load. When a copy of software is requested via the GPCIPL Menu Display, the program reads a table of values to determine if the load is available. If it is not available, a value of 'FFFF' is returned. If 'FFFF' is returned, the self-test program will annunciate the error, deschedule the load and return to the job scheduler. NOTE: This error can also be annunciated due to incorrect values present in the transaction, message or insert tables.	Incorrect keyboard item entry for a PASS or BFS load or incorrect GPC resident table builds.
DEU => ## ERASED **ERROR CODE: 212**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that the Display Buffer of the DEU specified by ## has been overwritten with alternating patterns of hex '5555' and 'AAAA', 100 times. This operation is initiated by entering an 'ITEM 27+1 EXEC', followed by 'ITEM 28 EXEC' on the GPCIPL Menu Display. The self-test program will annunciate the message and continue.	Not an error condition.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
ITEM 27 PURGE SELECTION NOT 1-8 **ERROR CODE: 213**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=the keystroke count R4=the invalid option number.	The CPU has determined that an invalid memory purge option was requested on the GPCIPL Menu Display or an incorrect number of keystrokes were supplied from the Display Electronics Unit (DEU). The GPCIPL Menu Display provides for 8 different memory operations via 'ITEM 27+X EXEC', where X equals 1-8 for a DEU erase, MMU1 overwrite, MMU1 erase, MMU1 checksum, MMU2 overwrite, MMU2 erase, MMU2 checksum and GPC erase, respectively. If the option number is not in the valid range or the keystroke count does not equal 6, the self-test program will ignore the request, annunciate the error and return to the job scheduler.	Failure of the DEU or an invalid option request for a GPC memory purge operation.
ITEM # 1-16 AND 27-29 NOT VALID IF PAGE 2 IS UP **ERROR CODE: 214**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R3=0001 0000.	The CPU has determined that an invalid item entry was requested on the GPCIPL Menu Display page 2. When page 2 is present on the CRT, only Items 17-26 are valid. The 'SYS SUMM' key is used to toggle the GPCIPL display page between 1 and 2. The self-test program will ignore the request, annunciate the error and return to the job scheduler.	An invalid item entry was requested from GPCIPL Menu page 2.
INVALID PAGE 2 REQUEST -> ITEM # 1-16 OR 27-29 IS UP **ERROR CODE: 215**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The CPU has detected an illegal request for a change to GPCIPL Menu Display page 2, via the 'SYS SUMM' key. When a PASS load, a BFS load or a memory purge operation is in progress (i.e. any of Items 1-16 or 27-29 currently selected), it is invalid to request a change in the GPCIPL Menu Display. The self-test program will ignore the request, annunciate the error and return to the job scheduler.	An invalid request for page 2 was requested during an active load or purge operation.
PURGE OPTION MUST BE SET BEFORE ITEM 28 EXEC **ERROR CODE: 216**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R6=the option number in its display format.	The CPU has detected an illegal procedure for initiating a memory purge operation from the GPCIPL Menu Display. The user has failed to correctly select an option, using an 'ITEM 27+X EXEC', prior to executing the 'ITEM 28 EXEC' to start the operation. Valid options are 1-8, for a DEU erase, MMU1 overwrite, MMU1 erase, MMU1 checksum, MMU2 overwrite, MMU2 erase, MMU2 checksum and GPC erase, respectively. The self-test program will ignore the request, annunciate the error and return to the job scheduler.	An incorrect procedure used to perform a memory purge operation.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
ITEM N OR ITEM N<N> ERROR => N MUST BE NUMERIC **ERROR CODE: 217**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R2=the invalid keystroke R4=the keystroke count.	The CPU has detected a non-numeric character in an 'ITEM X' or 'ITEM XY EXEC' command entered on the GPCIPL Menu Display. The characters must be between zero and nine to be valid. If not, the self-test program will ignore the request, annunciate the error and return to the job scheduler.	A non-numeric item number was specified for a keyboard item entry.
ITEM NUMBER < 1 AND > ## NOT SUPPORTED **ERROR CODE: 218**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=the item number entered.	The CPU has determined that an item number entered on the GPCIPL Menu Display is equal to '0' or is out of the valid range. The ## field will contain the highest possible item number available on the GPCIPL Menu Display. The self-test program will ignore the request, convert the maximum valid item number into a format which can be displayed on the CRT, insert the value into the error message, annunciate the error and return to the job scheduler.	An invalid item entry was requested from the GPCIPL Menu page.
PURGE OPTION IN PROGRESS => ## DESCHEDULED **ERROR CODE: 219**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that the CPU has descheduled either a GPC Self-test, a System Software Loader routine (SSL Check) or a Flight Software Loader routine (LOAD Check) after a memory purge operation was requested. The job(s), scheduled to run prior to the request for the purge operation, are removed from the job scheduling queue. The ## field within the message will contain 'LOAD CHK', 'SELF TEST' or 'SSL CHECK' to specify the operation name which was descheduled by the purge operation. The self-test program will annunciate the message and continue.	Not an error condition.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
ILLEGAL KYBD ENTRY WHILE ## IN PROGRESS **ERROR CODE: 220**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R4=item number entered.	The CPU has determined that an illegal keyboard entry has been made while a memory purge operation was in progress. 'ITEM 29 EXEC', which terminates the execution of the selected purge operation, is the only legal keyboard entry during execution of a purge operation. These operations are initiated using 'ITEM 27+X EXEC' and 'ITEM 28 EXEC', where X can equal 1-8 and ## will be replaced by: DEUERASE, MM1OVWRT, MM1ERASE, MM1CKSUM, MM2OVWRT, MM2ERASE, MM2CKSUM or GPCERASE. A 'PURGE COMPLETE' message will be annunciated when the operation terminates successfully. The self-test program will ignore the request, insert the appropriate text into the message, annunciate the error and return to the job scheduler.	Keyboard inputs other than 'ITEM 29 EXEC' were made on the GPCIPL Menu Display during a purge operation.
DEU KYBD BUFFER CHECKSUM ERROR **ERROR CODE: 221**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R0=the sum of the 16-halfword DEU poll response (incl. the checksum).	The checksum supplied by the Display Electronics Unit (DEU) in its 16-halfword poll response does not equal the checksum computed by the CPU. The DEU responds to a poll by the GPC with a poll response which contains display, keystroke and status information, along with a checksum. The CPU verifies the data in the response by adding each halfword together and storing it into General Register 0 (R0). The result should equal zero, ignoring overflow. If not, the self-test program will ignore the poll response, annunciate the error and return to the job scheduler.	Failure of the DEU or register R0.
MASS MEMORY 1 OR 2 NOT SELECTED **ERROR CODE: 222**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: R5=DIA register.	Neither the Mass Memory Unit 1 nor 2 (MMU1 or MMU2) has been selected to perform a Display Electronics Unit (DEU) or a default PASS load. Each time the DEU poll routine is executed, the CPU issues a Program Controlled Input (PCI) instruction to the IOP to read the Discrete Input A (DIA) register. When a DEU or default PASS load has been requested, bits 4 and 5 of the DIA register are checked to verify that one of them is set to '1'. This indicates that one MMU has been specified, via the IPL Source Select Switch, to exclusively perform the software load. The self-test program will annunciate the error and continue.	Failure or incorrect setting of the IPL Source Select Switch, failure of the DIA register or the Control/ Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
DEU => ## IS DE-ASSIGNED **ERROR CODE: 223**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that the GPC has successfully deassigned the Display Electronics Unit (DEU), specified by ##, using the keyboard command 'GPC/CRT 0Y EXEC'. The self-test program will convert the DEU identifier number into a format which can be displayed on the CRT, insert the value into the message, annunciate the message and continue.	Not an error condition.
DEU ## IS ASSIGNED TO GPC %% **ERROR CODE: 224**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that the GPC has successfully assigned the Display Electronics Unit (DEU), specified by ##, to the GPC specified by %%, using the keyboard command 'GPC/CRT XY EXEC'. The self-test program will convert the GPC and DEU identifier numbers into a format which can be displayed on the CRT, insert the values into the message, annunciate the message and continue.	Not an error condition.
## DEULOAD CHECKSUM ERROR IN TRANSACTION # # **ERROR CODE: 225**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a Display Electronics Unit (DEU) Control Program (DCP) load, a Critical Format load or a Self-Test Program load does not equal the checksum computed by the CPU. The CPU verifies the data in each block transferred by adding each halfword. The second ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The self-test program will attempt 2 retries before annunciating the error, then will terminate the job and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC or an incorrect checksum supplied on the MMU build.
## BSL AREA: CHECKSUM ERROR IN TRANSACTION: ## **ERROR CODE: 226**	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a Backup Flight System Loader (BSL) load does not equal the checksum computed by the CPU. The CPU verifies the data in each block transferred by adding each halfword. The second ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The self-test program will attempt 2 retries before annunciating the error, then will terminate the job and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC or an incorrect checksum supplied on the MMU build.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
## PURLOAD CHECKSUM ERROR IN TRANSACTION # # ERROR CODE: 227	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	The checksum of a block of data transferred from the GPC during a purge operation does not equal the expected value. The CPU verifies the data in each block transferred by adding each halfword. The second ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the operation failed. The self-test program will attempt 2 retries before annunciating the error, then will terminate the job and return to the job scheduler.	Possible transmission error during the transfer of data from the GPC during a purge operation.
## PURGE COMPLETE ERROR CODE: 228	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that the CPU has successfully completed 1 of 8 purge operations. These operations are initiated using 'ITEM 27+X EXEC' and 'ITEM 28 EXEC', where X can equal 1-8. The field ## in the message will be replaced by: DEUERASE, MM1OVWRT, MM1ERASE, MM1CKSUM, MM2OVWRT, MM2ERASE, MM2CKSUM or GPCERASE. The self-test program will annunciate the message and continue.	Not an error condition.
## PURGE STARTED ERROR CODE: 229	CRT Fault Message. Error Code listed on upper right of GPCIPL Menu Display pages. Significant Registers: none.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that the CPU has successfully initiated 1 of 8 purge operations. These operations are initiated using 'ITEM 27+X EXEC' and 'ITEM 28 EXEC', where X can equal 1-8. The field ## in the message will be replaced by: DEUERASE, MM1OVWRT, MM1ERASE, MM1CKSUM, MM2OVWRT, MM2ERASE, MM2CKSUM or GPCERASE. A 'PURGE COMPLETE' message will be annunciated when the operation terminates successfully. The self-test program will annunciate the message and continue.	Not an error condition.

APPENDIX B

APPENDIX

APPENDIX B
BACKUP SYSTEM LOADER (BSL) ERROR MESSAGES

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP MACH CHECK **ERROR CODE: 1**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected Machine Check (MC) Interrupt has been generated during execution of the Backup System Loader (BSL) for at least 1 of the following 6 reasons: 1) the IOP has detected even parity in an address specified while the CPU or IOP was addressing locations in GPC Extended Memory, memory locations '28000'-'33FFF' (int code '0001'); 2) the IOP has detected even parity in the data read from memory during a Direct Memory Access (DMA) 'Out' operation (int code '0002'); 3) the CPU has detected even parity stored in the contents of a CPU memory location (int code '0003'); 4) the IOP has detected even parity in the data read from GPC Extended Memory (int code '0004'); 5) the CPU has attempted to execute an instruction read from Read Only Storage (ROS) which contained even parity when output from the ROS Data Register (int code '0005'); 6) an interrupt code greater than '0005' or less than '0001' is contained in the MC Program Status Word (PSW) following an MC interrupt. Upon detection of the error, the CPU will generate a MC interrupt with the listed interrupt code, store it along with the current PSW into low core locations 40-43 and execute the MC interrupt handler. If the error occurs during a store operation, the CPU will inhibit the write to prevent an unwanted memory alteration. The BSL program will annunciate the error, load the current PSW with the address where the interrupt occurred and continue from that point.	Failure of the CPU, CPU main memory, the CPU ROS parity generator, the CPU ROS data register, the CPU interrupt microcode, IOP main memory or the IOP parity generator circuitry.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
EXT 0 INT-IOP FL **NOTE: Text is identical to Error Code 7.** **ERROR CODE: 4**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 0 (EX0) Interrupt has been generated with the Voter Fail Latch bit set in Interrupt Register A (INTA), indicating a critical failure of the computer. The IOP voter logic receives the outputs of the fail vote discrete receivers and sets the voter fail latch whenever 2 or more fail votes are received from other GPCs. The IOP will set INTA bit 1 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The Backup System Loader (BSL) program will annunciate the error and continue checking INTA for further error indications. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the CPU, the IOP or the R/M logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
EXT 0 INT-ROS PE **ERROR CODE: 6**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 0 (EX0) Interrupt has been generated with the IOP Read Only Storage (ROS) parity error bit set in Interrupt Register A (INTA), indicating that a parity error has been detected during an attempt by the IOP to execute a microinstruction. The microinstruction, read from ROS and placed in the ROS data register (ROSDR), contained bad (even) parity when output from the ROSDR. Upon detection of the parity error, the IOP will set INTA bit 3 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 interrupt handler. The Backup System Loader (BSL) program will annunciate the error and continue checking INTA for further error indications. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the CPU, the IOP IOP PROM or the ROS data register.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP EXT 0 INT **ERROR CODE: 9**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 0 (EX0) Interrupt has been generated with either invalid bit(s) set in Interrupt Register A (INTA) or no bits set at all. The value of INTA is set by the IOP following detection of an EX0 error condition, and is read by the CPU using a Program Controlled Input (PCI) instruction. The EX0 interrupt handler stores the register value into a General Purpose Register and checks to determine if any of bits 0-4 are set to '1'. If not, the Backup System Loader (BSL) program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the IOP, the Control/Monitor logic or INTA.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP EXT 3 INT **ERROR CODE: 1 1**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 3 (EX3) Interrupt has been generated during execution of the Backup System Loader (BSL). This priority of interrupt is not utilized in the AP-101B and should not occur during a software load. Upon detection of the error, the IOP will set the appropriate bit(s) in Interrupt Register D (INTD) to '1' and send the EX3 interrupt signal to the CPU via hardwire. The CPU will then generate the EX3 Interrupt, store the current Program Status Word (PSW) into low core locations 90-93 and execute the EX3 Interrupt handler. The BSL will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register D format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Zeroes (0).	Failure of the CPU, the IOP or INTD.
UNEXP EXT 4 INT **ERROR CODE: 1 2**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 4 (EX4) Interrupt has been generated during execution of the Backup System Loader (BSL). This priority of interrupt is not utilized in the AP-101B and should not occur during a software load. Upon detection of the error, the IOP will set the appropriate bit(s) in Interrupt Register E (INTE) to '1' and send the EX4 interrupt signal to the CPU via hardwire. The CPU will then generate the EX4 Interrupt, store the current Program Status Word (PSW) into low core locations 98-9B and execute the EX4 Interrupt handler. The BSL will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register E format: (0-no interrupt, 1-interrupt) Bits 0-3=Spare Bits 4-31=Undefined.	Failure of the CPU, the IOP or INTE.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP PROG CHECK ERROR CODE: 1 4	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected Program Check (PC) Interrupt has been generated during the Backup System Loader (BSL) for at least 1 of the following 10 reasons: 1) the CPU attempted to execute a Privileged instruction while in the Problem State (int code '0001'); 2) an invalid memory location was specified during a CPU memory access (int code '0003'); 3) the result of a fixed point operation was too large to be stored in a fullword format (int code '0004'); 4) a floating point operation resulted in a fractional value equal to zero, but a non-zero characteristic (int code '0005'); 5) a floating point divide instruction was attempted using unnormalized inputs (int code '0006'); 6) the CPU attempted a store data into a protected memory location (int code '0007'); 7) a floating point operation resulted in a characteristic that could not be expressed in a 7-bit format (int code '0009'); 8) a Convert to Floating Point (CVFL) instruction resulted in a non-zero fraction and a characteristic greater than 127 (int code '000A'); 9) a floating point operation other than CVFL resulted in a non-zero fraction and a characteristic greater than 127 (int code '000B'); 10) a floating point divide instruction was attempted using a denominator equal to '0' (int code '000C'). Upon detection of the error, the CPU will generate a PC interrupt with the listed interrupt code, store it along with the current Program Status Word (PSW) into low core locations 48-4B and execute the PC Interrupt handler. The BSL will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure to place the CPU in Supervisor State prior to execution of a privileged instruction, the CPU, incorrect address inputs to a memory access operation, a fixed point register, incorrect fixed point operands, a floating point register, incorrect floating point operands.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
INVALID SVC # **ERROR CODE: 17**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	A Supervisor Call (SVC) Interrupt has been generated with an illegal interrupt code in the Supervisor Call Program Status Word (PSW). Interrupt code > the number of allowable SVC's in the Backup System Loader (BSL) The SVC interrupt logic is utilized by the Backup System Loader (BSL) program to annunciate any error detected in the CPU, the IOP or main memory. Upon detection of the error, the CPU will generate an SVC interrupt with an interrupt code equal to the error code, store it along with the current Program Status Word (PSW) into low core locations 58-5B and execute the SVC Interrupt handler. The BSL program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU.
BSL RR.VV.P.II.MM LOADED **ERROR CODE: 18**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates which version of the Backup System Loader (BSL) is loaded, where: RR = Revision number, VV = Version number, P = Patchset number, I I = I-load set number, MM = mass memory area number. A copy of the BSL is loaded into GPC memory in response to a request for a Backup Flight Software (BFS) load, via an Item 2,4 or 6 on the GPCIPL Menu Display. The BSL program will annunciate the message, set a flag so that the message is only annunciated one time and continue.	Not an error condition.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
DEU BITE ERROR **ERROR CODE: 2 1**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The Display Electronics Unit (DEU) has detected a critical BITE error. The Backup System Loader (BSL) , during the DEU Poll Response routine, has found bit 14 of the DEU poll response headerword to be set to '1' (critical BITE error present). The routine will test the Software Status Register for a power transient. If a power transient occurred (bit 2 = '1'), the BITE error is assumed to have been induced and control is passed back to the job scheduler. Otherwise, the BSL program will save the contents of DEU BITE status registers 1 and 2, annunciate the error and return to the job scheduler.	Failure of the DEU

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU I/O ERROR BCE 18-19 **ERROR CODE: 23**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The CPU has detected an IOP BCE I/O error during either of 2 cases: 1. during the Mass Memory Unit (MMU) scheduler routine to verify the status of the MMU and BCEs prior to initiating an I/O transaction, or 2. during a routine which performs a checksum of each block of data transferred from the MMU to the GPC. The program reads the Go/No-Go Register using a Program Controlled Input (PCI) instruction, then checks bits 18 and 19 (MMU BCEs) for a value of '1'. If any equal '0', indicating that the necessary BCE is in a No-Go state, the Backup System Loader (BSL) will annunciate the error, reset the MMU load environment for a possible restart and continue. Go/No-Go Register format: (0-NoGo, 1-Go) Bit 0=MSC Bits 1-24=BCEs 1-24, respectively Bit 25=self-test processor Bits 26-31=Undefined.	Failure of a BCE or the MIA.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU SOURCE SELECT SW POS. ERROR **ERROR CODE: 2 5**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The CPU has determined that the Mass Memory Unit (MMU) 1 and 2 IPL source select discretes in the Discrete Input A (DIA) register indicate an incorrect configuration of the MMU BCEs for a Backup Flight Software (BFS) load. The CPU issues a Program Controlled Input (PCI) instruction to read DIA register in the IOP. A check is then made to verify that one of the IPL source select discretes (bit 4-MMU1, bit 5-MMU2) is set to '1'. If both are set or neither is set, the Backup System Loader (BSL) will deschedule the job, reset the MMU load environment for a possible restart, annunciate the error and return to the job scheduler. DIA Register format: Bit 0=Halt Bit 1=Standby Bit 2=Run Bit 3=IPL Bit 4=MMU 1 IPL Bit 5=MMU 2 IPL Bit 6=MMU 1 Ready Bit 7=MMU 2 Ready Bit 12=Inhibit Channels 10-13 Bit 13=Inhibit Channels 14-17 and 20-23	Failure of the IPL Source Select Switch, the Control/Monitor logic, the DIA register or an incorrect setting of the IPL Source Select Switch.
MMU EXPECTED POSITION ERROR **ERROR CODE: 2 6**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The CPU has determined that either Mass Memory Unit 1 or 2 is positioned incorrectly during BFS software load. The CPU checks the MMU's current File-Track-Subfile position against the position it has computed following each position and read transaction. A check is then made to verify that the values agree. If not, the Backup System Loader (BSL) will reset the MMU load environment for a possible restart, annunciate the error and return to the job scheduler.	Failure of the MMU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
TFL FORMAT=>%% -COMPUTED CHECKSUM ERROR- XACTION ## **ERROR CODE: 3 0**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a Telemetry Formal Load (TFL) does not equal the checksum computed by the CPU. The CPU verifies the data in each block transferred by adding each halfword. The field ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The %% field will contain the TFL format number which was being loaded when the error occurred. This error can occur due to different data transfer rates between MMUs. The Backup System Loader (BSL) will retry the load until the checksums of 4 consecutive blocks miscompare, then will terminate the job, reset the MMU load environment for a possible restart and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC, an incorrect checksum supplied on the MMU build or different data transfer rates between MMUs (SMS B06546A).
TFL FORMAT=>0102 EXPCTD/IS FORMAT %% =>XACTN ## **ERROR CODE: 3 1**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a Telemetry Formal Load (TFL) does not equal the checksum computed by the CPU. The TFL format being loaded was '0102'. The CPU verifies the data in each block transferred by adding each halfword. The field ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The Backup System Loader (BSL) will terminate the job, reset the MMU load environment for a possible restart and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC or an incorrect checksum supplied on the MMU build.
TFL FORMAT=>0105 EXPCTD/IS FORMAT %% =>XACTN ## **ERROR CODE: 3 2**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a Telemetry Formal Load (TFL) does not equal the checksum computed by the CPU. The TFL format being loaded was '0105'. The CPU verifies the data in each block transferred by adding each halfword. The field ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The Backup System Loader (BSL) will terminate the job, reset the MMU load environment for a possible restart and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC or an incorrect checksum supplied on the MMU build.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
KSC RR.VV.P.II.MM LOADED_SET MMU SEL SW TO OFF **ERROR CODE: 27**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates which version of Backup Flight Software (BFS) is loaded, where: RR = Revision number, VV = Version number, P = Patchset number, I I = I-load set number, MM = mass memory area number. The Backup System Loader (BSL) will annunciate the message, set a flag to allow 1 annunciation and continue.	Not an error condition.
BFS CHECKSUM ERROR XACTION ## **ERROR CODE: 28**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a Backup Flight Software (BFS) load does not equal the checksum computed by the CPU. The CPU verifies the data in each block transferred by adding each halfword. The field ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The Backup System Loader (BSL) will terminate the job, reset the MMU load environment for a possible restart and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC or an incorrect checksum supplied on the MMU build.
IMU CHKPT CHEKSUM ERR-XACTION ## **ERROR CODE: 29**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The checksum of a block of data supplied by the Mass Memory Unit (MMU) during a IMU Checkpoint Data load does not equal the checksum computed by the CPU. The CPU verifies the data in each block transferred by adding each halfword. The field ## in the message will contain the transaction count at the time the error occurred, to aid in determining the block of data where the load failed. The Backup System Loader (BSL) will terminate the job, reset the MMU load environment for a possible restart and return to the job scheduler.	Possible transmission error during the transfer of data from the MMU to the GPC or an incorrect checksum supplied on the MMU build.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU WILL NOT GO READY **ERROR CODE: 2 4**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The CPU has determined that the Mass Memory Unit (MMU) 1 and 2 ready discretes in the Discrete Input A (DIA) register do not indicate that either MMU is available for use. The discretes are statused prior to initiating an MMU operation. The CPU issues a Program Controlled Input (PCI) instruction to read DIA register in the IOP. A check is then made to verify that the ready discrete of the selected MMU (bit 6-MMU1, bit 7-MMU2) is set to '1'. The CPU allows 90 seconds for the MMU to set the discrete following its selection via the IPL Source Select Switch. If the time expires before the discrete is set, the Backup System Loader (BSL) will deschedule the job, reset the MMU load environment for a possible restart, annunciate the error and return to the job scheduler. DIA Register format: Bit 0=Halt Bit 1=Standby Bit 2=Run Bit 3=IPL Bit 4=MMU 1 IPL Bit 5=MMU 2 IPL Bit 6=MMU 1 Ready Bit 7=MMU 2 Ready Bit 12=Inhibit Channels 10-13 Bit 13=Inhibit Channels 14-17 and 20-23	Failure of the MMU or the Control/Monitor logic.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
MMU BITE STATUS REG A OR B ERROR **ERROR CODE: 2 2**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The CPU has detected a Mass Memory Unit (MMU) 1 or 2 BITE error during an MMU I/O transaction. The CPU reads MMU Status Registers A and B and checks for any bits to be set. If so, the Backup System Loader (BSL) will reset the MMU load environment for a possible restart, annunciate the error and return to the job scheduler on the second occurrence of the error. MMU Status Register A format: (0 - no failure, 1 - failure) Bit 0=Power transient Bit 1=File address not equal Bit 2=Command error Bit 3=Write protect violation Bit 4=Invalid Op code MMU Status Register B format: (0 - no failure, 1 - failure) Bit 0=Data Count word low Bit 1=Read tape data dropout Bit 2=read chk assurance lost Bit 3=Read tape Parity Error Bit 4=MIA invalid Manchester Code Bit 5=Command Rec'd/not ready Bit 6=EOF Block Count zero Bit 7=No search address compare Bit 8=Malfunction Bit 9=BOT sensed Bit 10=EOF sensed Bit 11=power supply out of tolerance-operation aborted Bit 12=MIA bit count error Bit 13=MIA parity error Bit 14=Invalid 101 check Bit 15=MIA data/address error	Failure of the MMU, the MIA or incorrect inputs to the MMU transaction.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP IOP PG INT **ERROR CODE: 1 9**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 2 (EX2) Interrupt has been generated while the Backup System Loader (BSL) was running, indicating that an MSC program has unexpectedly executed a programmable Interrupt (@INT) instruction. The EX2 class of interrupts is designed for any MSC program defined function. Prior to execution of the @INT instruction, the MSC program will set the appropriate bit(s) in Interrupt Register C (INTC). The IOP will then send the EX2 interrupt signal to the CPU via hardwire. The CPU will generate the EX2 interrupt, store the current Program Status Word (PSW) into memory locations 88-8B and execute the EX2 Interrupt handler. The BSL will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register C format: (0=no interrupt, 1-interrupt) Bits 0-11=MSC Program Interrupts 1-12, respectively Bits 12-31=Zeroes (0).	Failure of the IOP or an MSC program.
NO DEU RESPONSE **ERROR CODE: 2 0**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	The Display Electronics Unit (DEU) has not responded to a poll request by the Backup System Loader (BSL). The DEU is polled two times per second, and its response is checked on the minor cycles immediately following those in which it is polled (7 and 20). A flag is set whenever a response has been received by the GPC. The BSL program will annunciate the error and return from the poll request routine to the job scheduler.	Failure of the DEU hardware.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP SVC INT ERROR CODE: 15	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	A Supervisor Call (SVC) Interrupt has been generated with an illegal interrupt code in the Supervisor Call Program Status Word (PSW). The interrupt code is not in the range of 1 through 32, the current number of SVCs in the Backup System Loader (BSL) program. The SVC interrupt logic is utilized by the program to annunciate any error detected while the BFS load is being performed. Upon detection of the error, the CPU will generate an SVC interrupt with an interrupt code equal to the error code, store it along with the current PSW into low core locations 58-5B and execute the SVC Interrupt handler. The BSL program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU.
SVC INT -WILD BR ERROR CODE: 16	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	A Supervisor Call (SVC) Interrupt has been generated with an illegal interrupt code in the Supervisor Call Program Status Word (PSW). The interrupt code is equal to 'C9FB', which is the hexadecimal pattern loaded into unused GPC core at GPC initialization. A SVC interrupt code of this value indicates a wild branch, or a branch out of the Backup System Loader (BSL) program. The SVC interrupt logic is utilized by the program to annunciate any error detected while the BFS load in being performed. Upon detection of the error, the CPU will generate an SVC interrupt with an interrupt code equal to the error code, store it along with the current PSW into low core locations 58-5B and execute the SVC Interrupt handler. The BSL program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP IM INT ERROR CODE: 1 3	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected Instruction Monitor (IM) Interrupt has been generated due to an attempt by the CPU to fetch the contents of an unprotected memory location for use as program code. All program code or instructions must be stored as protected data in memory, and the instruction monitor feature prevents a program from erroneously executing data as code. The CPU, upon detection of the violation, will store the current Program Status Word (PSW) into low core locations 70-73 and execute the IM Interrupt handler. The Backup System Loader (BSL) will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point.	Failure of the CPU or incorrect store protect bit setting within GPC memory.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP EXT 1 INT **ERROR CODE: 1 0**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 1 (EX1) Interrupt has been generated. If the interrupt code in the EX1 Program Status Word (PSW) is non-zero, it indicates that the CPU or AGE interface detected the error. In this case, the interrupt can be due to any of 6 reasons: 1) even parity was been detected in the data response to a Programmed Controlled Input (PCI) instruction (int code '0001'); 2) even parity has been detected in the data supplied during a Direct Memory Access (DMA) 'In' operation (int code '0002'); 3) an invalid address has been specified during a DMA operation (int code '0003); 4) a store protect violation has occurred during a DMA 'In' operation (int code '0004); 5) even parity has detected in the address supplied during a Direct Memory Access (DMA) operation (int code '0005'); 6) the AGE interface has been activated (int code '0006'). An interrupt code of '0000' indicates that the IOP detected the error. Interrupt Register B (INTB) must be interrogated to determine the type of error detected. In either case, the CPU will generate the EX1 interrupt, store the current PSW into low core locations 80-83 and execute the EX1 Interrupt handler. The BSL program will annunciate the error, load the current PSW with the address stored at the time of the interrupt and continue from that point. Interrupt Register B format: (0-no interrupt, 1-interrupt) Bit 0=PCI/PCO Parity error Bit 1=DMA Instruction Parity error Bit 2=DMA Data Read Parity error Bit 3=Burst DMA Word Count Excess Bit 4=Q Overflow Bit 5=DMA Timeout Bits 6-31=Undefined.	Failure of the IOP.

APPENDIX C
INDEX

+28 V dc Detector Circuit, 8-12
+ 5 V dc Detector Circuit, 8-12
1.04 Hz Transfer Data, 14-15

-A-

Access Time Error Signature, 5-70
ACCUMULATOR TIME OUTPUTS, 6-13
ACCUMULATORS, 6-12
ANNUNCIATION SYSTEMS, 13-30
Antenna Management, 13-53, 16-8
APPLICATIONS SOFTWARE, 5-38, 12-2, 13-44, 11.2-1
APU Fuel Quantity Computation, 3-50, 16-8
Assignments CRT/DEU, 4-43, 90, 103

-B-

BCE 18 or 19 MMU I/O, 5-66
    27 MMU I/O, 5-65
    Programs, 5-36, 11.1-14, 17
BFC CRT DISPLAY/SELECT 4-103, 112, 117, 119
    CRT Select Logic, 8-9
    CRT Switch Positions, 4-91
    HARDWARE, 8-3
BFS Engage, 8-38, 42, 50, 11.1-30
    SM DISPLAYS, 16-13
    SM FUNCTIONAL REQUIREMENTS, 16-2
    SM MONITORING, 16-17
    Disengage, 8-41
    TMBU, 13-47, 16-18
BITE GPC MESSAGE, 2-42
    CPU, 2-51
    DEU, 4-17
    DU PHOSPHOR PROTECT, 4-20
    DU STATUS, 4-21
    SCREEN INFORMATION, 4-29
    DEU POWER SUPPLY, 4-39
    DEU/DU INTERFACE, 4-44
    ERROR CHECK CONTROL, 4-46
    MCDS BIT POSITION, 4-50, 57
    DEU STATUS WORD, 4-62, 65
    DEU STATUS INFORMATION, 4-71, 73
    OTP ITEM B, 4-82
    CRT, 4-87
    DEU HDR BUFFER, 4-132
    MMU SELF FAIL, 5-20
    MMU 1/2 HAL NAME, 5-22
    MTU OSCILLATORS, 6-8
    FREQ. DIVIDERS, 6-11
    REGISTER FF MDM, 6-13
    REGISTER OF MDM, 6-14
    IRIG CHANNEL FAIL, 6-15
    MTU UPDATE LOGIC, 6-19
    MDM, 7-22
    MDM RESPONSE WORDS, 7-4
    MDM BSR, 7-11
    MDM BSR FILE, 7-25, 26, 27
    MDM CORE POWER SUPPLY, 7-22, 26
    MDM A/D, IOM, PROM, BSR, 7-27
    READ INPUTS FOR GPC, 7-43, 44, 45
    PASS/BFS I/O PROC, 8-44
    D/L WORDS (FLOW CHART), 11.1-8
    DEU LOADER,11.3-4
    MTU Bit 1, 6-22
    MTU Bit 12, 6-23
    MTU Bit 13/14/15, 6-24
    MTU Bit 16, 6-24
    MTU Bit, 2/3 6-22
    MTU Bit 4, 6-22
    MTU Bit 5/6, 6-22
    MTU Bit 7/8, 6-23
    MTU Bit 9/10/11, 6-23
    MTU STATUS REGISTERS, 6-20
    MMU Status Words, 5-21
    MTU WORD, 6-22
Bootstrap Loader, 9-4, 11.1-24
BUFFER FILL MCDS MSG4, 4-52
Buffer Poll Pass, 13-53
BYPASS CONDITIONS, 12-7

-C-

Panel C2, 4-5
Cal Curve, Flight Software, 13-44, 45
CAM Interface, 2-77
Channel Control, 2-57
Command Processing, 5-32
Commands and Responses, 5-17
Compare operations, 5-57
Completion Monitoring, 5-35
Component Locations AV BAY/
  Flight Deck, 2-1, 4-7
Console Tools, 4-90
Control Bus Down Impacts, 8-40, 8-42
    Monitor IOP, 2-64
    Sequence MMU TCP, 5-57
COOLING GPC, 2-4
    KEYBOARD, 4-10
    DU, 4-22, 23
    DEU, 4-30
    MTU, 6-1
    OTB/PTB, 6-25
    MDM, 7-35
CPU Microprogram, 2-31
Crew Controls, 2-4, 4-4, 5
Crew Text, 4-121, 5-64
MTU CRITICAL FAILURE MODES, 6-33
CRT Power, 4-6
    Select, 4-6, 11.1-30, 11.3-5, 6
CRT/DEU, 4-90, 93, 100

-D-

D/L Parameters, 4-129
DATA BUS NETWORK, 2-1, 56, 69, 7-4, 12-4
Data Formats, 2-31
DDU POWER, 8-1
    INTERFACE HCED/RHC, 8-18, 19
    POWER LOSS IMPACTS, 8-41
IPL Default or Menu Option, 9-6
DEU Connectors, 4-34
    Control Program, 4-45, 58, 121
    MEMORY DUMP, 4-49, 56
    Physical Layout, 4-33
    to DU Analog Signals, 4-19
DEU/DU Interface, 4-44
DEU/IOP Interface, 4-44
DEU/KYBD Interface, 4-43
Direct I/O, 2-30
    Memory Access (DMA), 2-28, 30, 11.1.4-6

DISCRETE INPUT/OUTPUT, 2-71
DISENGAGE OPERATIONS, 8-41
DISPLAY DATA FILL, 4-54, 86
Electronics Unit, 4-1, 2, 27, 32, 11-23, 27
Unit, 4-1, 2, 17, 27, 39, 8-18
Unit Connectors, 4-24
DK Transmitter Enables, 4-90
Downlink Software, 11.2.2-1
DPS Interface (Chart) MCDS/PASS, 4-3
Dump Downlist, 11.2.3-2
Formats (Software/HISAM), 11.2.3-3, 11.1.3-3
operations MMU, 5-57

-E-

ENGAGE OPERATIONS, 8-36
/Disengage Logic, 8-21
/Disengage Transition, 14-14
Erroneous MTU Output, 6-36
Error Log Reset, 11.1.2-1, 2
Completion Processing, 5-37
EVENT TIMERS, 6-31, 11.3-16
Extended Addressing, 2-50

-F-

Fail Vote Discretes, 2-77
Failure Detection, 6-28

-F-

Fail Vote Discretes, 2-77
Failure Detection, 6-28
Fault Alarm Avoidance, 16-5
Annunciation, 16-5, 16-11-16-12
Detection, 2-51
Detection and Annunciation, 13-5, 13-30, 16-2, 16-11
FCOS I/O ERRORS, 12-12
FDA PDT, 13-30
TIMING CONCERNS, 13-39
BFS, 16-3
Data Formats, 13-31
Data Structure, 13-33
Processing, 13-36
FF MDMs, 6-13, 14, 17, 20
Flex MDM, 12-6, 13-23
FORMAT DATA FILL, 4-55, 4-66, 11.3-5
Fuel Cell Computation, 13-50
Cell Purge Sequence, 13-51

-G-

G9 One Shot Transfer Data, 14-16
General Purpose Registers, 2-33
GPC DATA BUS CONTROL, 8-44
Discrete Input, 2-80
HALT OPERATIONS, 8-32
HALT Relay, 8-9
Initialization, 11.1-22-11.1-24, 11.3-2-11.3-3, 11.3.3-1
INTERNAL OPERATIONS, 12-1
Memory Spec, 11.2.3-1
Mode Discrete Inputs, 2-81
POLL, 4-49
Select Logic, 8-15
SELECT OPERATIONS, 8-33
MCDS Data Communications, 4-48
IPL, 2-85, 8-31, 9-4, 9-6-9-7, 11.1-23-11.1-24, 11.3.3-1
IPL DEU Characteristics, 8-31

-H-

HALT Discrete Failure, 2-83
Hand Control Engage Driver, 8-18
Controller Engage Driver, 8-38
HARDWARE PHASE, 9-1
Hardware Response, 9-1
STATUS WORD 1, 4-63
STATUS WORD 2, 4-65

High Rate Inputs, 13-20
HISAM DUMP, 11.1.3-4, 5
Hydraulic Fluid Temp Control, 13-51
Water Boiler, 13-52, 16-8

-I-

I/O ERROR CONDITIONS, 12-8
RESET, 4-122, 7-46, 8-40, 49, 50, 12-7, 11, 13, 13-24, 26, 11.1.2-2, 14-8, 10, 11
IFM PROCEDURES, 4-46, 8-37, 49
Inner Data Flow (CPU), 2-34
INPUT/OUTPUT PROCESSOR, 2-56, 58, 7-1, 11.1-27
Interrange Instrumentation Group B, 6-14
Interrupts, 2-37, 40, 11.1-8, 14-21
IOP Input Discrete Hardware Commonality, 2-75
MIA, 2-69, 4-27, 14-4
Microcode, 9-1, 9-3
IPL BITE STATUS REQUEST, 4-49, 51
HISAM Dump, 11.1.3-1
Initialization, 11.3.3-1
IUS GNC Transfer, 13-55

-K-

Keyboard Layout, 4-12
Processing, 8-32
Unit, 4-1, 2
Unit Connectors, 4-13

-L-

Limit Sensing, 16-4, 11
LISTEN COMMAND, 8-44, 46, 49, 14-4
Local Store, 2-33
Low Rate Inputs, 13-18

-M-

MAJ FUNC, 4-6, 4-29
Manual Ops Transition, 8-27
Mass Memory Directory, 4-121, 5-50
MCC Special Comps, 4-133
MCDS OVERVIEW, 4-1
INTERFACE (CHART), 4-3
KEYBOARD (CHART), 4-12
CRT TALKBACK, 4-21
MSG 5 (TIME FILL), 4-53
MEMORY DUMP, 4-57
POLL RESPONSE, 4-58
DEU STAT WORDS, 4-62
SASTP, 4-77
KB SYNTAX, 4-98
PROBLEMS, 4-120
PARAMETERS, 4-129
BFS PARAMETERS, 4-131
PASS/BFS INSTRUCTION PROCESS, 8-30
Displays, 4-67
OPERATIONS, 4-48
Physical Characteristics, 4-7
STATUS REQUEST, 4-49, 50

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
UNEXP CLOCK2 INT **ERROR CODE: 2**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected Program Counter (Clock) 2 Interrupt has been generated while the Backup System Loader (BSL) was executing. The CPU maintains 2 clocks, each composed of one 16-bit binary hardware register and one 16-bit main memory location. The hardware register, with a clock time of 1 microsecond, will cause the associated memory location to be decremented by 1 each time the register passes from '0000' to 'FFFF' (approx. every 65.536 microseconds). When the main memory location counts down to zero, a Clock 2 Interrupt will be generated, the CPU will store the contents of the current Program Status Word (PSW) into low core locations 68-6B and the Clock 2 Interrupt handler will be executed. Clock 2 is used to verify the presence of computer control. Each time the clocks are initialized, clock 2 is set to a value .5 microseconds greater than Clock 1. A loss of computer control is signified by Clock 2 expiring before clock 1. The BSL will annunciate the error, load the current PSW with the return address stored at the time of the interrupt and continue from that point.	Failure of the CPU.
EXT 0 INT-W/D **ERROR CODE: 3**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 0 (EX0) Interrupt has been generated with the Watchdog (Go/No-Go) Timer bit set in Interrupt Register A (INTA), indicating a cyclic failure of the computer. The Watchdog timer must be periodically set by the CPU, using Program Controlled Output (PCO) instructions, to function properly. The IOP, upon detection that the timer has counted to its maximum value, will set INTA bit 0 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The Backup System Loader (BSL) program will annunciate the error and continue checking INTA for further error indications. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the CPU or IOP, the C/M logic or Redundancy Management (R/M) logic.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1988—561-009/80567

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
EXT 0 INT-IOP FL **NOTE: Text is identical to Error Code 4.** **ERROR CODE: 7**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 0 (EX0) Interrupt has been generated with the IOP Fault bit set in Interrupt Register A (INTA), indicating that the IOP oscillator has stopped. The oscillator, contained in the Processor Element logic of the IOP, provides the timing necessary for the proper time-sharing of MSC and BCE functions. Upon detection that the oscillator has stopped, the IOP will set INTA bit 4 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The Backup System Loader (BSL) program will annunciate the error and continue checking INTA for further error indications. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the IOP, the Processor Element logic or the IOP oscillator.
END-EXT 0 ERRORS **ERROR CODE: 8**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	This message DOES NOT indicate an error condition, but is an advisory message for information purposes, only. Indicates that all bits within Interrupt Register A (INTA) have been interrogated, and there are no further External 0 (EX0) interrupts to annunciate. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Not an error condition.

ERROR CONDITION	HOW MANIFESTED TO USER	ERROR DESCRIPTION	POSSIBLE CAUSES
EXT 0 INT-C/M ID **ERROR CODE: 5**	CRT Fault Message with current error code listed on upper right of BSL2IPL Menu Display page.	An unexpected External 0 (EX0) Interrupt has been generated with the Control/Monitor (C/M) Idle bit set in Interrupt Register A (INTA), indicating the C/M logic is available for operations. The C/M logic, which is controlled by the CPU using Program Controlled Output (PCO) instructions, controls and monitors such functions of the IOP as MIA transmit and receive enable, IOP discrete output, BCE halt/proceed status, interrupt register status and interruption of the CPU. The C/M Idle interrupt should only occur when a Master Reset IOP or IOP Fail Reset has been performed or at GPC power on. The IOP will set INTA bit 2 to '1' and send the EX0 interrupt signal to the CPU via hardwire. The CPU will then generate the EX0 interrupt, store the current Program Status Word (PSW) into low core locations 78-7B and execute the EX0 Interrupt handler. The Backup System Loader (BSL) program will annunciate the error and continue checking INTA for further error indications. Interrupt Register A format: (0-no interrupt, 1-interrupt) Bit 0=Watchdog timer timeout Bit 1=IOP Fail latch Bit 2=C/M Idle (reset to "0" before the check for this error) Bit 3=IOP ROS parity error Bit 4=IOP Fault (Oscillator stopped) Bit 5=Spare Bits 6-31=Undefined.	Failure of the CPU, the IOP or C/M logic.