

## Space Flight Operations Contract

---

# DPS Familiarization Workbook DPS FAM 21002

**Mission Operations Directorate  
Space Flight Training Division  
Flight Training Branch**

February 23, 2006

Final Version

This document has been reviewed and updated.  
No subsequent updates to this document are anticipated or required due to the  
approaching shuttle program closure.

Contract NAS9-20000

Copyright © 2004 by United Space Alliance, LLC. These materials are sponsored by the National Aeronautics and Space Administration under Contract NAS9-20000. The U.S. Government retains a paid-up, nonexclusive, irrevocable worldwide license in such materials to reproduce, prepare derivative works, distribute copies to the public, and to perform publicly and display publicly, by or on behalf of the U.S. Government. All other rights are reserved by the copyright owner.



**DPS Familiarization Workbook  
DPS FAM 21002**

Prepared by

*Original approval obtained*

---

Carla Sugar  
Shuttle DPS and Navigation Group

Approved by

*Original approval obtained*

---

Walter J. Barnett  
Manager, Shuttle DPS and Navigation Group

Mission Operations Directorate  
Space Flight Training Division  
Computer and Navigation Training Branch

## INTRODUCTION

The DPS Familiarization 21002 workbook (this document) and the three DPS FAM lessons (taken in the Single System Trainers [SST]) together comprise the DPS Familiarization Learning Module. The purpose of this module is to provide the student with a basic understanding of the operation of the Shuttle Data Processing System, as well as the rationale behind the operating procedures.

The DPS Familiarization 21002 Workbook provides the theory and rationale behind the various concepts presented in the SST DPS FAM lessons. The workbook should be read prior to taking the SST DPS FAM lessons. The workbook sections do not correspond to the order of the material in the SST DPS FAM lessons. Instead, the material is divided into sections based on related concepts.

The DPS FAM 21006 and DPS FAM 22006 lessons are, in effect, two parts of one long lesson. The material is tied to a timeline sequence from pre-launch through entry. While the presentation will be associated with a timeline sequence, it will not be slaved to it. There is a 30-minute prebrief associated with each of these lessons. In addition, each prebrief will contain an assessment of the student's knowledge prior to taking the lesson. If the student is insufficiently prepared for a given lesson, the lesson in question will be rescheduled and remedial training will be suggested. In addition, the DPS FAM 22006 lesson will provide opportunities for assessment of student understanding/performance prior to scheduling DPS FAM 23006. The purpose of these two SST lessons is to exercise the basic operations for DPS in the context of flight-related events. No Flight Data File (FDF) procedures will be used in these two lessons; however, the instructor will provide procedures specific to the lessons, as needed. Certain malfunctions will be used during these two lessons; however, the purpose is only to illustrate the concepts being presented, as these are familiarization lessons, not malfunction recovery procedures lessons.

The DPS FAM 23006 lesson is a nominal-operations familiarization lesson, closely following the nominal timeline for a shuttle flight. It will begin in the pre-launch timeframe and the student will perform all nominal DPS FDF procedures in sequence. The rationale for the structure of the procedures will be reinforced in context with the mission timeline.



**DPS Familiarization Workbook  
DPS FAM 21002**

LIST OF EFFECTIVE PAGES

The current status of all pages in this document is as shown below:

<u>Page No.</u>	<u>Change No.</u>
Revision Log	Rev. C
Preface	Rev. C
List Of Effective Pages	Rev. C
i – iv	Rev. C
1-1 – 1-7	Rev. C
2-1 – 2-3	Rev. C
3-1 – 3-3	Rev. C
4-1 – 4-4	Rev. C
5-1 – 5-2	Rev. C
6-1 – 6-2	Rev. C
7-1 – 7-3	Rev. C
8-1 – 8-8	Rev. C
9-1 – 9-11	Rev. C
10-1 – 10-5	Rev. C
11-1 – 11-2	Rev. C
12-1 – 12-3	Rev. C
13-1 – 13-10	Rev. C
A-1 – A-3	Rev. C

## **PREFACE**

The contents of this document was provided by the Space Shuttle DPS/Navigation Group, Computer and Navigation Training Branch, Space Flight Training Division, Mission Operations Directorate (MOD), Lyndon B. Johnson Space Center (JSC), and National Aeronautics and Space Administration (NASA). Technical documentation support was provided by Integrated Documentation Services (IDS) and Hernandez Engineering, Inc. Any questions concerning this training manual or any recommendations should be directed to the responsible department; DT35/DPS/Navigation.

This material is for training purposes only and should not be used as a source of operational data. All numerical data, displays, and checklist references are intended as examples. To determine any prerequisites before using this document, consult the applicable Certification Plan. For shuttle manuals, consult the Flight Operations Support Personnel Training Guide (Blue Book) or the Crew Training Catalog. For Space Station manuals, consult the appropriate Space Station Certification Training Guide or Training Administration Management System (TAMS). The applicable training package should be studied before attending any classroom session or lesson for which this is a prerequisite.

A Training Materials Evaluation is included at the end of this document. Inputs on this sheet will be used to evaluate the lesson material. You do not need to sign the sheet.

## CONTENTS

Section	Page
<b>1.0 DPS HARDWARE CONTROLS .....</b>	<b>1-1</b>
1.1 PANEL O6 .....	1-1
1.1.1 GPC Power Switch.....	1-1
1.1.2 OUTPUT Switch and Talkback .....	1-2
1.1.3 MODE Switch and Talkback .....	1-3
1.2 IPL HARDWARE CONTROLS .....	1-4
1.2.1 Details Of "Sleep" Mode.....	1-5
1.3 MULTIPLEXER/DEMULTIPLEXER POWER SWITCHES .....	1-6
1.4 VISUAL CUES .....	1-6
1.4.1 PANELS O1 AND F7 .....	1-6
<b>2.0 SOFTWARE OVERVIEW.....</b>	<b>2-1</b>
2.1 PASS VS. BFS.....	2-1
2.2 PASS SOFTWARE STRUCTURE .....	2-1
<b>3.0 OPS TRANSITIONS.....</b>	<b>3-1</b>
3.1 GPC PROCESSES DURING AN OPS TRANSITION.....	3-1
3.2 GPC RECONFIGURATION WITHOUT AN OPS TRANSITION .....	3-3
3.3 THE G3 ARCHIVE .....	3-3
<b>4.0 GPC INITIAL PROGRAM LOAD .....</b>	<b>4-1</b>
4.1 OVERVIEW.....	4-1
4.2 DEFAULT OPTION IPL.....	4-1
4.3 MENU OPTION IPL (PASS) .....	4-3
4.4 BFS IPL.....	4-4
<b>5.0 DATA BUSES .....</b>	<b>5-1</b>
<b>6.0 PORT MODING.....</b>	<b>6-1</b>
<b>7.0 BFS TRACKING .....</b>	<b>7-1</b>
<b>8.0 DK BUS ASSIGNMENT .....</b>	<b>8-1</b>
8.1 OVERVIEW.....	8-1
8.2 BFC CRT DISPLAY AND SELECT SWITCHES .....	8-2
8.3 MAJ FUNC SWITCHES .....	8-3
8.4 GPC/CRT KEY.....	8-6
8.5 OPS TRANSITIONS/MODE RECALLS .....	8-7
8.6 SPECIAL CRT ASSIGNMENT RULES .....	8-8

<b>Section</b>	<b>Page</b>
<b>9.0 MULTIFUNCTION ELECTRONIC DISPLAY SYSTEM.....</b>	<b>9-1</b>
9.1 OVERVIEW.....	9-1
9.2 MEDS DATABUS NETWORK.....	9-2
9.3 MEDS HARDWARE CONTROLS .....	9-4
9.3.1 MDU Controls .....	9-4
9.3.2 IDP/CRT Controls .....	9-4
9.3.3 Other Controls.....	9-5
9.4 MEDS SOFTWARE .....	9-5
9.4.1 Overview .....	9-5
9.4.2 Edgekey Menu System .....	9-5
9.4.3 MEDS Generic Screen Format .....	9-6
9.4.4 MEDS Displays .....	9-8
9.5 OPERATIONS .....	9-11
9.5.1 MDU Configuration.....	9-11
<b>10.0 DOWNLIST AND UPLINK .....</b>	<b>10-1</b>
10.1 PULSE CODE MODULATION MASTER UNIT .....	10-1
10.2 TELEMETRY FORMAT LOADS .....	10-2
10.3 GPC DOWNLIST .....	10-2
10.4 VARIABLE PARAMETERS.....	10-3
10.5 UPLINK.....	10-4
<b>11.0 CHECKPOINTS .....</b>	<b>11-1</b>
11.1 CHECKPOINT INITIATE.....	11-1
11.2 CHECKPOINT RETRIEVE.....	11-1
<b>12.0 MASS MEMORY INPUT/OUTPUT.....</b>	<b>12-1</b>
12.1 DESCRIPTION .....	12-1
12.2 MMU INSIGHT.....	12-1
12.3 ROLL-IN SPECS.....	12-2
<b>13.0 BFS ENGAGE.....</b>	<b>13-1</b>
13.1 BFS ENGAGE OVERVIEW .....	13-1
13.2 BFS ENGAGE PROCESS .....	13-2
13.2.1 Hardware .....	13-2
13.2.2 Software.....	13-8
13.2.3 Failures .....	13-9
13.3 BFS DISENGAGE PROCESS .....	13-10
 <b>Appendix</b>	
<b>A ACRONYMS AND ABBREVIATIONS .....</b>	<b>A-1</b>



## FIGURES

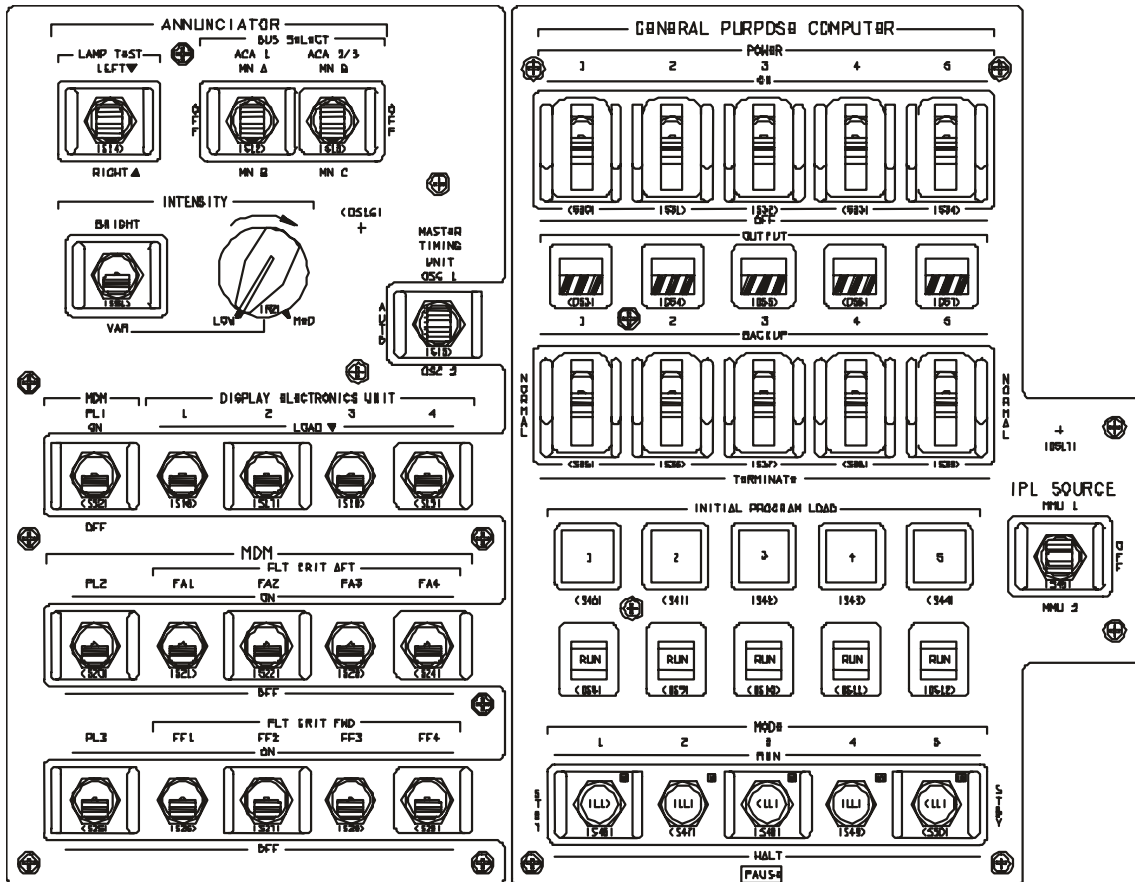
<b>Figure</b>	<b>Page</b>
1-1 Panel O6 .....	1-1
1-2 Talkback indications associated with GPC MODE switch positions (arrows show direction of switch motion) .....	1-4
1-3 SPEC 001 DPS UTILITY .....	1-5
1-4 Computer annunciation matrix .....	1-6
1-5 Caution and Warning Matrix on Panel F7 .....	1-7
2-1 Orbiter flight computer software .....	2-2
2-2 GPC Memory Configurations .....	2-3
3-1 SPEC 006 GPC/BUS STATUS .....	3-2
4-1 GPC IPL MENU display .....	4-4
5-1 GPC data buses .....	5-2
6-1 MDM port connections on a string .....	6-1
6-2 MDM internal redundancy .....	6-2
7-1 The BFS listen process .....	7-2
8-1 DK data buses .....	8-1
8-2 BFC CRT switches on Panel C3 .....	8-3
8-3 IDP/CRT controls and keyboards on Panel C2 .....	8-3
8-4 GPC/BUS STATUS display (SPEC 6) .....	8-5
9-1 MDUs on Panels F6, F7, and F8 .....	9-1
9-2 MEDS Data Bus Network .....	9-3
9-3 Multifunction Display Unit .....	9-4
9-4 MEDS Edgekey Menu Hierarchy .....	9-6
9-5 MEDS Generic Screen Format .....	9-7
9-6 COMP ADI/HSI Flight Instrument Display .....	9-8
9-7 OMS/MPS Subsystem Status Display .....	9-9
9-8 MEDS Maintenance Display .....	9-11
10-1 Downlink flow through the PCMMU .....	10-1
10-2 GPC downlist path through the PCMMU .....	10-4
10-3 Uplink flow through the DPS .....	10-5

<b>Figure</b>		<b>Page</b>
11-1	SM checkpoint data flow.....	11-2
12-1	SPEC 1, DPS UTILITY .....	12-2
13-1	BFS engage pushbutton interface to the BFS modules.....	13-3
13-2	BFC input/output overview.....	13-4
13-3	BFC engage logic.....	13-5

## 1.0 DPS HARDWARE CONTROLS

### 1.1 PANEL O6

The hardware controls for the GPCs and MDMs are located overhead of the commander's seat on Panel O6, as shown in Figure 1-1. Each group of switches, pushbuttons, and talkbacks is discussed below.



TD6010101. PNL, 1

Figure 1-1. Panel O6

#### 1.1.1 GPC Power Switch

The POWER switch is located in the farthest aft row of GPC switches on Panel O6. When ON, power is supplied from the 3 main electrical buses, providing 3 redundant power sources for each GPC.

GPCs have volatile memories. Except under extreme circumstances, procedures do not call to power off a GPC. GPCs do "sleep" to conserve power. The "sleep" mode is described in more detail later.

### 1.1.2 OUTPUT Switch and Talkback

The OUTPUT switch is not read by a GPC directly. It is read by an intervening device called the Backup Flight Controller (BFC). The primary purpose of the OUTPUT switch is to provide hardware control of a GPC's ability to transmit over flight critical (FC) data buses. This is accomplished via a discrete called the I/O TERM B. This discrete is generated by the logic circuits in the BFC and sent to the GPC. When the I/O TERM B is high, or "1", the GPC does not have the ability to transmit over the FC buses. When the I/O TERM B is low, or "0", the GPC does have the ability to transmit over the FC buses (even though it may not necessarily be doing so at the moment).

The switch has three positions: NORMAL, TERMINATE, and BACKUP. In NORMAL, the GPC has the hardware ability to transmit over the flight criticals (I/O TERM B low), while the TERMINATE position inhibits this ability (I/O TERM B high). Since the GNC GPCs are responsible for flying the orbiter, they have their OUTPUT switches in NORMAL. On orbit, the SM GPC has its OUTPUT switch in TERMINATE to provide a hardware guarantee, since it is not supposed to command anything on the flight critical buses. The BACKUP position is used to tell the BFC logic circuits which GPC is loaded with the Backup Flight System (BFS) software. This GPC is the one that will take control of the vehicle in the event of a BFS engage.

The BFC consists of six hardware modules that provide the following four basic DPS functions:

- a. Routing of the GPC HALT discrete
- b. Enabling the crew selection of a DPS display for either BFS usage or GPC IPL
- c. Enabling or inhibiting a GPC's ability to transmit over the FC data buses (via the OUTPUT switch)
- d. Implementing the BFS engage and disengage processes

When the BFS OUTPUT switch is in BACKUP, its transmission over the flight criticals is inhibited unless the BFS is engaged. Should the crew engage the BFS, the BFC gives control of the vehicle to the BFS.

The OUTPUT talkback (tb) reflects the GPC's ability to transmit over FC buses. While influenced by the OUTPUT switch, this tb is not controlled by the OUTPUT switch. A gray tb indicates the ability to transmit, while a barberpole tb (a diagonal black and white striped pattern) indicates that the ability to transmit is inhibited. Nominally, PASS is controlling the orbiter and BFS is not engaged. The GNC GPCs will have gray tb's, and the BFS will have a barberpole tb. If the BFS is engaged, the PASS tb's will go to barberpole (bp) while the BFS GPC (the one now in control) will have a gray tb.

In summary, the OUTPUT switch, by means of a discrete called the I/O TERM B, either enables or inhibits the capability of the GPCs to transmit over the flight criticals. If the switch is in NORMAL, then the GPCs have the hardware ability to transmit over flight criticals. If the OUTPUT switch is in TERMINATE, the GPC is hardware inhibited from transmitting over the flight criticals, regardless of what the software demands.

### 1.1.3 MODE Switch and Talkback

The purpose of the MODE switch is to control a GPC's software processing state. The transitions to the various positions (RUN, STBY, and HALT) of the MODE switch generate different responses in PASS and BFS. The MODE tb provides feedback to the crew by indicating RUN, bp, or IPL.

#### 1.1.3.1 PASS

In the RUN position the GPC supports normal software processing (systems and applications software). The GPC will reflect RUN on its MODE tb. When a GPC is moded from RUN to STBY, the GPC stops processing PASS software and performs an orderly shutdown. The GPC is still in a software controlled state and is monitoring the movement of the MODE switch to see if it is moded to RUN. A bp tb reflects completion of the orderly shutdown. The OUTPUT tb also goes to bp because the GPC is no longer capable of transmitting over the FC buses. Placing the MODE switch in HALT puts the GPC in a hardware controlled state, where no software processing occurs. The hardware (firmware) monitors a few switches, including the MODE switch and the IPL controls (discussed below). The MODE tb remains bp. Therefore, whenever a GPC is taken from RUN to HALT, it is necessary to pause in STBY and wait for the tb to go from RUN to bp. This ensures that the GPC has performed its orderly shutdown and that, barring any failures, the GPC can be reactivated by moding the GPC back to RUN.

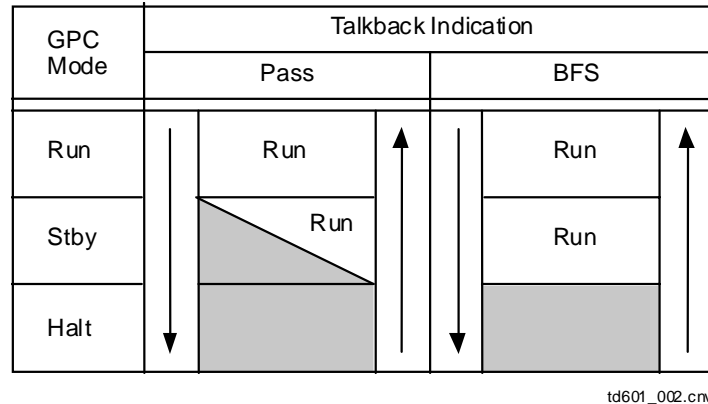
When the MODE switch is moved from HALT to STBY, the GPC goes into a software controlled state and is reinitialized. The MODE tb changes from bp to RUN when this process is complete (in order to let the user know that the GPC is healthy and processing software now). It is important to pause in STBY before going to RUN in order to allow the reinitialization process to be completed. Even though the tb indicates RUN, PASS software is not executing in STBY. A small program that monitors the MODE switch and performs a few other functions is running instead. Moding from STBY to RUN will start PASS system software (OPS 000) processing and the MODE tb will remain RUN. The OUTPUT tb will go from bp to gray. Subsequent OPS transitions are required to initiate any applications software processing. (Note: No other actions should be performed by the crew from ten seconds before until ten seconds after MODE'ing any PASS GPC to RUN, since doing so can interfere with that GPC joining the common set. The GPC could fail-to-sync then or at a subsequent OPS transition. The worst-case scenario is that the entire common set could be corrupted. Due to these concerns, the general rule is: ***Everyone should be "hands off" during GPC initializations.***)

### 1.1.3.2 BFS

BFS is not like PASS. Full processing of the BFS software occurs whether the MODE switch is in either RUN or STBY; the difference is that in STBY, the BFS cannot command the PL data buses pre-engaged. During ascent and entry, all 4 PASS GPCs are processing GNC. The BFS is performing Systems Management (SM) functions, such as antenna management, over the PL data buses. (Of course, the BFS is also monitoring PASS GNC functions, so it will be ready to take over control of the vehicle if engaged.)

Once on orbit, placing the BFS in STBY allows the SM GPC to command over the PL data buses. Whether in RUN or STBY, the BFS GPC's MODE tb will remain in RUN. When the BFS is later taken to HALT, the BFS GPC enters the hardware controlled state and behaves the same as any GPC in HALT, with its tb's showing bp.

Figure 1-2 provides a summary of GPC MODE switch and tb states for PASS and BFS. Note that the tb pattern seen in PASS depends upon the direction the mode switch is moved.



**Figure 1-2. Talkback indications associated with GPC MODE switch positions (arrows show direction of switch motion)**

## 1.2 IPL HARDWARE CONTROLS

An Initial Program Load, or IPL, is the process where a GPC is loaded from the Modular Memory Units (MMUs) with either PASS systems software or all of BFS's software. To perform this process, two IPL hardware controls are used: the IPL SOURCE SELECT Switch and an IPL pushbutton for each GPC.

The IPL SOURCE switch is used to reserve an MMU for IPL transactions. Either MMU 1 or MMU 2 may be used, although MMU1 is normally used. While an MMU is selected for an IPL that MMU cannot be used for other transactions such as OPS transitions. It is therefore important to return this switch to OFF when not performing an IPL.

There is an interface with PASS software that forces the GPCs to disregard the position of the IPL SOURCE switch. This interface is on SPEC 1, the IPL SOURCE SW MASK selection (ITEM 38). When this selection is enabled, the software will not look at the IPL SOURCE switch position and therefore will not reserve that MMU for the IPL process. This allows an MMU to be used if there is a failure of the switch. When this selection is disabled, the GPCs will respect the position of the switch and not use the chosen MMU for any transactions. Normally, when IPLing a GPC, we disable this mask so an active GPC will not access an MMU while another GPC is being IPL'ed.

```

2011/001/      DPS UTILITY      4 125/16# 04# 32
                                03/11# 04# 32

MMU ASSIGN     PORT ASSIGN
  1  2
GNC 1* 2      STRING 1 15* 16
SM 3* 4        2 17* 19
PL 5* 6        3 19* 20
DPS 0 7* 8     4 21* 22
               PL 1/2 23* 24

MMU STATUS
  1 RDY
  2 RDY

MMU SOURCE/BLIS 25 7795
GPC/MMU 9*      26 879A
MMU MMU 1D      27 77EB
GPC/LDB 11      28 77EF
               29 8010
               30 007A
CKPT RBT RV     31 2444
  BNA 12*       32 3302
               33 3400
RTC             34 3401
  13 _____
  CMD 14 _____

UL CNTL
AUTO 35*
BNA 36
INH 37

IPL SOURCE SW
MASK 38*

G3 ARCHIVE
LOAD 4B
RBT RIBVE 49*

GROUND OPS
G5B POLL BNA 50
SM G5B INH 51
SM C/O BNA 52
GNC C/O BNA 53
BFC C/O BNA 54
ALT PL9 TB 55

```

TD6010121. CRT: 1

**Figure 1-3. SPEC 001 DPS UTILITY**

The IPL pushbutton is used to initiate the IPL process for a given GPC. More details will be discussed in the IPL section of this workbook.

### 1.2.1 Details Of "Sleep" Mode

A GPC is put into a "sleep" mode by placing the GPC's MODE switch in HALT, keeping its POWER switch ON, and having the IPL SOURCE switch OFF. After a short period of time (about 4.5 minutes), if the GPC detects no action (such as MODE or IPL SOURCE switch movement), the GPC will enter the sleep state. In this sleep state, the GPC will use minimum power, just enough to maintain its memory contents. The GPCs have a small, internal auxiliary processor that cyclically checks the contents of the GPC's memory for errors and corrects them. It is able to do this since each memory location contains extra bits of error correcting code (ECC). Movement of the MODE switch will "wake up" the GPC. Each GPC contains a rechargeable Battery Backup Unit (BBU) to preserve the contents of the GPC's memory in the event the GPC loses power; however, the BBU will not support error correction processing. The BBU charges whenever power is applied to the GPC.

### 1.3 MULTIPLEXER/DEMUTIPLEXER POWER SWITCHES

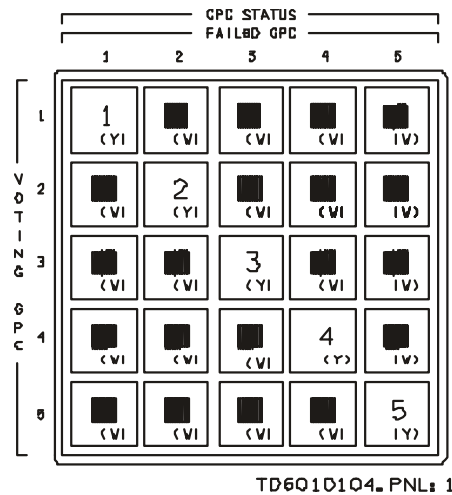
Multiplexer/Demultiplexers (MDMs) are hardware units that act as translators between the GPCs and the various equipment with which a GPC communicates. MDMs are named for the types of data they handle and their physical location in the orbiter. For instance, MDMs which format commands from GPCs to equipment critical to flight such as jets and aerosurfaces are called flight critical MDMs. These can be located in the forward or the aft area of the orbiter; if they are in the forward, they are flight forward (FF) MDMs; in the aft, they are flight aft (FA) MDMs.

Power switches for the MDMs are on Panel O6, to the left of the GPC control switches (refer to Figure 1-1). Each switch, with the exception of PL3, provides redundant power from two main buses. The PL3 MDM does not exist, so this is an unused switch.

### 1.4 VISUAL CUES

#### 1.4.1 PANELS O1 AND F7

These panels have lights that give insights to failures. The two portions of these panels that the DPS interfaces with are the computer annunciation matrix (CAM, see Figure 1-5) and the caution and warning matrix on Panel F7 (see Figure 1-5.)



**Figure 1-4. Computer annunciation matrix**

The CAM is a block of 25 lights, each GPC having a light or "vote" against the others. When a GPC finds a problem with another GPC, the GPC detecting the problem will issue a U-fail vote. These U-fails light the off-diagonal lights. When a GPC receives two or more U-fails, that GPC lights its own I-fail light on the diagonal (which inhibits the "bad" GPC from voting against the "good" ones). These votes are latched in RUN and HALT. This means that even if the condition that caused the votes is rectified, the votes will still be lit. There are three ways to clear votes: perform an OPS transition, perform an error log reset (via ITEM entry), or mode the GPC into STBY. This third method will clear the GPCs I-fail if there are less than two fail votes against that GPC. Moding to STBY will clear the CAM diagonal for a GPC that was in HALT, after the votes from the other GPCs have been cleared with one of the first two methods.



When an I-fail light is lit on the CAM, the 'GPC' light will be illuminated on the F7 caution and warning matrix. The 'Backup C/W' light will be illuminated when any class 2 failure is detected by GPC caution and warning software. Failure of a GPC is a class 2 failure.

O <sub>2</sub> PRESS	H <sub>2</sub> PRESS	FUEL CELL REAC (R)	FUEL CELL STACK TEMP	FUEL CELL PUMP
CABIN ATM (R)	O <sub>2</sub> HEATER TEMP	MAIN BUS UNDERVOLT (R)	AC VOLTAGE	AC OVERLOAD
FRON LOOP	AV BAY/ CABIN AIR	IMU	FWD RCS (R)	RCS JBT
H <sub>2</sub> O LOOP	RGA/ACCBL	AIR DATA (R)	LFT RCS (R)	RIGHT RCS (R)
————	LFT RHC (R)	RIGHT/AFT RHC (R)	LFT OMS (R)	RIGHT OMS (R)
PAYLOAD WARNING (R)	GPC	FCS (R) SATURATION	DMS KIT	OMS TVC (R)
PAYLOAD CAUTION	PRIMARY C/W	FCS CHANNEL	MPS (R)	————
BACKUP C/W ALARM (R)	APU TEMP	APU OVERSPBD	APU UNDERSPBD	HYD PRESS

( 34V73A7A2 )

TDB010103.PNL, 1

**Figure 1-5. Caution and Warning Matrix on Panel F7**

## **2.0 SOFTWARE OVERVIEW**

### **2.1 PASS VS. BFS**

You have already learned about the differences between PASS and BFS in the DPS OV 21002 workbook. Basically, BFS is a single string system that monitors PASS. BFS will control the orbiter upon crew action (BFS engage), if the PASS is not able to do so. BFS contains all the software it will ever need for a safe ascent, deorbit, and entry, without ever having to access the modular memory units. PASS, on the other hand, is a redundant, detailed system of four GPCs. This detail requires more software, so the MMUs must be accessed for different phases of flight, due to limited GPC memory.

### **2.2 PASS SOFTWARE STRUCTURE**

Recall that PASS software consists of two types: systems software and applications software. Systems software runs the GPC. It roughly corresponds to the operating system on a desktop computer. System software is responsible for tasks such as GPC to GPC communication, loading software from Modular Memory Units, and timekeeping activities. Applications software is software that runs the orbiter. This includes software that calculates orbiter trajectories and maneuvers, monitors various orbiter systems (such as power, communications, and life support), and supports mission specific payload operations.

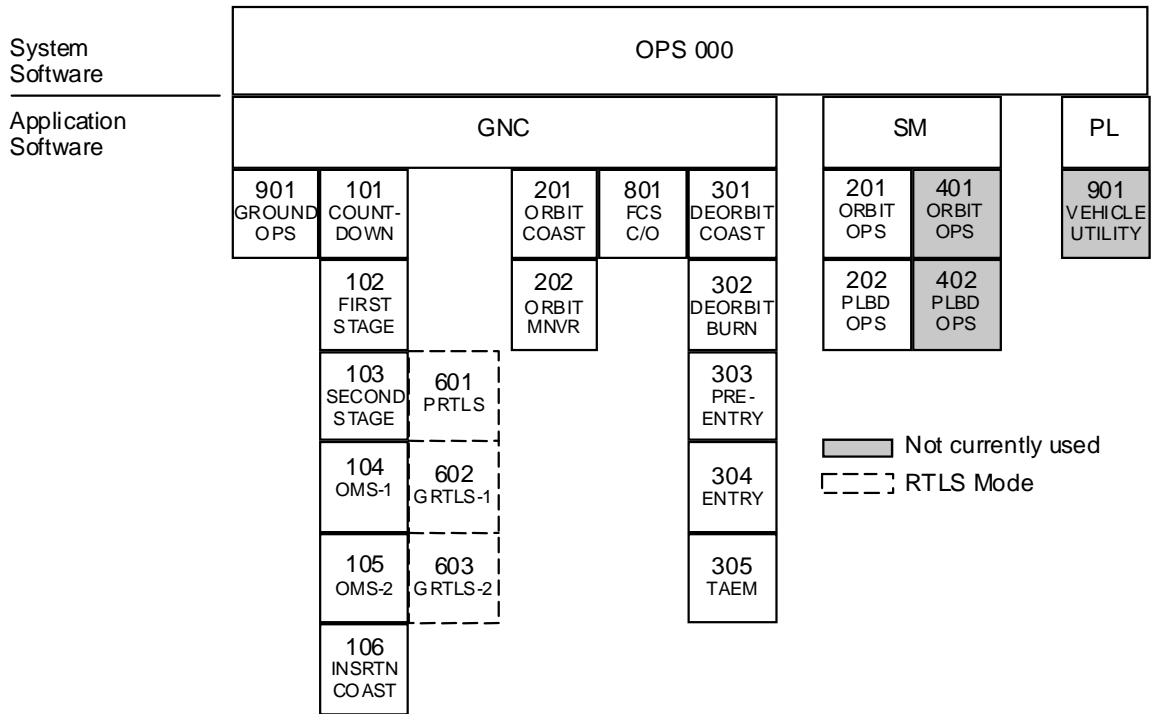
The applications software is divided into broad functional areas called Major Functions (MF). There are three major functions: Guidance, Navigation, and Control (GNC), which is the software for flight control and navigation; Systems Management (SM), which is the software for environmental management, limit sensing, and payload functions; and payload (PL), which is unfortunately misnamed because it does not support payload functions. PL major function software is used to change MMU contents at KSC and is not normally used during a flight.

Major Functions consist of smaller packages of software called Operational Sequences (OPS). An OPS is software used for a particular phase of flight. For example, OPS 1 is used for ascent. A GPC can hold only one OPS package at a time, with the exception of OPS 1 and 6. These two packages reside in the memory concurrently, in case OPS 6 software is needed for an RTLS abort. Getting software from the MMU takes extra time, and has the potential for failures.

Finally, each OPS is comprised of one or more Major Modes (MM) that address individual events in or subphases of the flight. The complete software structure is reflected graphically in Figure 2-1.

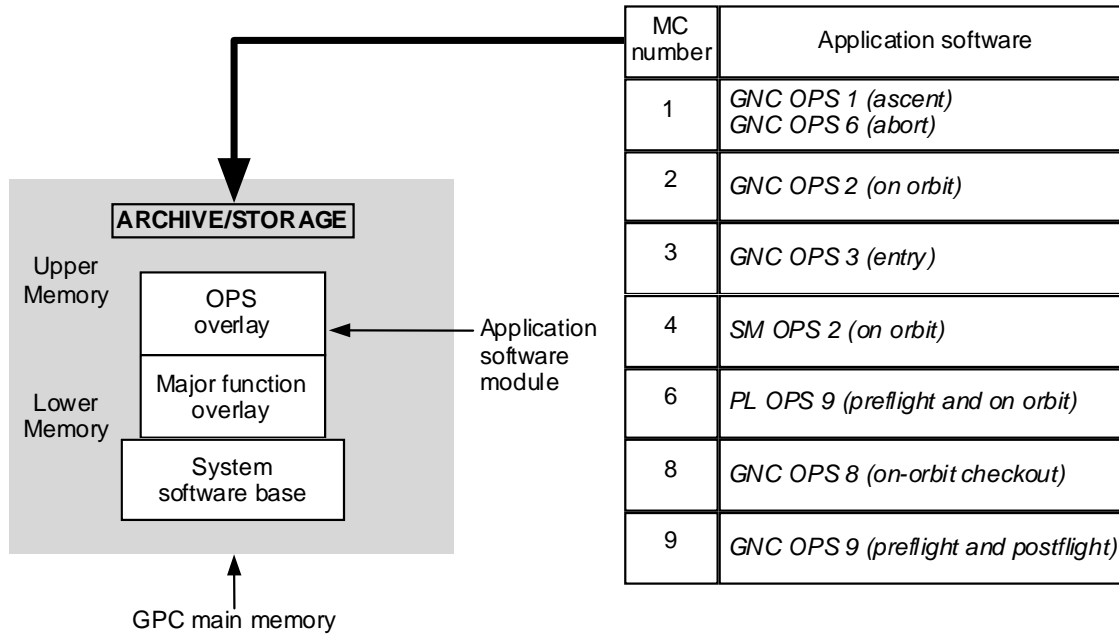
As mentioned earlier, it is possible for a GPC to only process its system software. It may or may not have applications software loaded in its memory. A GPC in this state is said to be in OPS 000. (Note that when a PASS GPC is taken from STBY to RUN, it is in OPS 000). However, it is not possible to process applications software without systems software.

The combination of systems and applications software in a GPC is referred to as a Memory Configuration (MC). Each Memory Configuration is assigned a number. These numbers are used to refer to software packages when the NBAT is configured. There are 8 PASS Memory Configurations. The GNC MC numbers are easy to remember: the number of the MC is the same as the GNC OPS that it contains (e.g., MC 2 contains GNC OPS 2).



td601\_006.cnv

**Figure 2-1. Orbiter flight computer software**



td601\_a\_007.cnv

**Figure 2-2. GPC Memory Configurations**

As can be seen from Figure 2-2, a memory configuration consists of a systems software base and an applications software module. The applications software module consists of two components: a major function base and an OPS overlay. The major function base contains the software and data that are common to all OPS within a given major function. An example of such data for the GNC MF is the orbiter's state vector; i.e., the orbiter's position and velocity. While different OPS use different calculation schemes to determine the state vector, it is constantly maintained by the GNC applications software in the GNC major function base in all phases of flight. The OPS overlay consists of the specific flight phase software (e.g., the OPS and Major Modes [OPS 1 Ascent: MM 101 - engine ignition, MM 102 - SRB powered flight, and so forth]). During OPS transitions, if a new major function base is needed, it will be loaded in first, followed by the requested OPS overlay.

In general, active software resides in the lower memory of the GPC. For PASS GPCs, the upper memory is used primarily for storage of the GNC OPS 3 (G3) archive. Having the G3 archive in upper memory allows for quick access to entry software in the event of a Transoceanic Abort Landing (TAL) and ensures that entry software is always available on-orbit.

### 3.0 OPS TRANSITIONS

#### 3.1 GPC PROCESSES DURING AN OPS TRANSITION

Since the GPCs have such limited memory, they must be reconfigured at different times during flight to process the correct software needed at that time. If certain rules are not met and checks are not passed, the GPC(s) will not load the requested software. This is why the three requirements for an OPS transition must be met: the major function switch must be in the correct position, the request must be made to a targeted GPC, and the transition must be legal. Understanding the steps a GPC goes through to load software can help you understand and remember the necessary conditions for an OPS transition.

1. The transition must be legal. This "legality" is based on a logic tree written in system software. Sometimes, system software is referred to as the flight computer operating system (FCOS); they are the same. Basically, legality means the transition must make sense. For example, it would not make sense to transition from OPS 201 (orbit software) to OPS 101 (ascent software). Therefore, this is illegal, and attempting it would result in an "ILLEGAL ENTRY" message. This legality check is referred to as "step 0" because the GPC has not actually begun the OPS transition steps yet; it has just checked to see if the transition is reasonable.
2. The GPC takes inventory of software available. The sources of GPC software in order of preference are a GPC's lower memory, a GPC's archive (OPS 3), or the MMUs. The OPS package and major function base are considered separately, so they can come from different sources. A good example of this is the transition from GNC OPS 201 to GNC OPS 301 for entry. A new OPS package is needed (OPS 3) but the currently processing GNC GPCs have the most current major function base, with necessary information such as the state vector and IMU data. The major function base will come from a currently processing GPC and the OPS overlay will come from the upper memory G3 archive.
3. The GPCs will check the MMUs, if needed. If the GPCs determine they need to use the MMUs to obtain the software, the GPCs, as designated on the current NBAT, will preposition the tapes to the correct location and load the needed software. The GPCs reference SPEC 1, DPS UTILITY, to determine which MMU should be tried first. Crew members must ensure that MMU selections on this SPEC are correct, since the GPCs use this information.
4. The GPCs stop processing the old OPS software.

5. The GPCs enter OPS 0. The time between old OPS and new OPS processing, when the new OPS is being transferred in, is a temporary state of OPS 0. It is sometimes called "pseudo OPS 0" because it is temporary. From the time of the OPS request through the completion of the load, no inputs or switch throws should be done. The GPCs are in transition doing critical checks and processing. It is essential that these processes not be interrupted. Remember: **Everyone should be "hands off" during OPS transitions.**
6. The GPCs implement the new bus assignments. This is per the target NBAT set up prior to the OPS transition. Since the GPCs configure to these assignments, crew members should pay special attention to these entries.
7. The GPCs get the new software. This will be from the MMUs or the GPCs, depending on what was determined earlier through the inventory check and SPEC 1.
8. The GPCs start new OPS processing. The completion of the load can be confirmed on the OPS page, SPEC 0, and SPEC 6.

		GPC/BUS STATUS					2 213/17: 16: 37 000/18: 16: 37	
		1	2	3	4	5		
		RUN	RUN	HALT	RUN	HALT		
		GZ	GZ	0	SZ	0		
MODE	GPC							
OPS								
STRING	1	FF	*				↓	
		FA	*				↓	
	2	FF		*			↓	
		FA		*			↓	
	3	FF	*				↓	
		FA	*				↓	
	4	FF		*			↓	
		FA		*			↓	
PL	1		↓			*		
	2		↓			*		
LAUNCH	1		↓			*		
	2		↓				↓	
CRT	1	*					↓	
	2		*				↓	
	3		↓				↓	
	4		↓			*		

TD6010903. CRT. !

**Figure 3-1. SPEC 006 GPC/BUS STATUS**

### 3.2 GPC RECONFIGURATION WITHOUT AN OPS TRANSITION

Sometimes it is necessary to have a GPC in OPS 0, for instance, when making a freeze-dried GPC. One can command an "OPS 000 PRO" to a GPC, but if it is a GNC GPC, all GPCs in the redundant set will go to OPS 0. *Whatever you do to one GPC in the redundant set, you do to all of them.* A single RS GPC can be forced into OPS 0 (and thereby removed from the redundant set) by cycling its MODE switch from RUN to STBY and back to RUN (remember to wait ten seconds before going from STBY back to RUN). Any PASS machine moded from STBY to RUN will be processing only OPS 0.

At other times, a GPC must be added to or deleted from the current redundant set. One may need to put a GPC to "sleep" to conserve power or may need to add one to the set for redundancy at a critical time. This is called contraction or expansion of the redundant set and *can only be performed in an 'X01' major mode.* If the target set of the NBAT is changed and the NBAT is invoked (via OPS/mode recall), the expansion or contraction will take place. Other changes in the NBAT (i.e., bus reassignments), can be performed with an OPS/mode recall in any major mode. (Note: The redundant set cannot be expanded or contracted in the special cases of major modes 101 or 601.)

### 3.3 THE G3 ARCHIVE

As is already understood, the archive is an OPS 3 overlay and resides in each PASS GPC's upper memory. The crew controls for loading and retrieving the archives are on SPEC 1. The LOAD item is enabled automatically when the first GPC is IPL'ed. When the GPCs transition from OPS 9 to OPS 1, the OPS 3 overlay will be loaded into upper memory and checked. The OPS 1 software will then be loaded. If any of the GPCs do not successfully load the OPS 3 overlay in the upper memory, the RETRIEVE item will not be enabled automatically, and the shuttle is not "go" for launch. If the loads are successful, the RETRIEVE item is automatically enabled. With the RETRIEVE enabled, the source of OPS 3 software will be the archive. Once the archive is loaded and the RETRIEVE item is enabled, the status on SPEC 1 will not change, even if problems in the upper memory develop. The crew has no insight into the status of the upper memory, but the MCC does. If there is a problem with upper memory, the option exists to inhibit RETRIEVE and use the MMUs as the source of software. This would only be done if the ground requests it.

The in-flight OPS transition from OPS 2 to OPS 3 nominally uses the G3 archives in the upper memory of the redundant set GPCs participating in the transition. When the GPC(s) perform the steps of an OPS transition (as above), the "inventory" step finds all needed software already in the GPCs. The GNC major function base in each GPC is retained, and the G3 software is copied from upper memory to lower memory. This is called a "fast transition", since it only takes about six seconds to complete.

## **4.0 GPC INITIAL PROGRAM LOAD**

### **4.1 OVERVIEW**

The purpose of the initial program load (IPL) process is to load each GPC with its operating software. This process initializes a GPC that would otherwise be simply a black box incapable of performing any vehicle control functions. After the IPL process is complete, we have a useful GPC that is fully tested and ready to perform work. There are two types of IPL processes available to load a GPC, the default option and the menu option. Both options give you a working machine, capable of supporting application software.

The IPL process itself is divided into two basic phases. The first phase is a hardware phase. This phase is implemented by a series of switch throws and button pushes which begin the IPL process. The second phase is implemented by software execution in the GPC itself. The IPL process is normally a procedure that ground personnel at KSC would do before a flight for each GPC. A crew member would not normally IPL a GPC during flight, unless the machine failed for some reason. (Over a dozen IPLs have been performed in-flight.)

Each MMU contains three copies of the PASS and two copies of the BFS software. The default option IPL loads the first copy of PASS systems software into a GPC. Any copy may be selected via the menu option IPL. Since the default option is for PASS only, the BFS must be loaded via the menu option IPL. Additionally, once the IPL is complete, applications software (major function bases and OPS overlays) as well as all other software used by that GPC, comes from the same copy that was used for the IPL. (Note: It is possible to change the source copy for OPS overlays for future transitions at any time during a flight by using ITEMS 52 - 54 on SPEC 0, GPC MEMORY, just in case some part of the software in the source copy is subsequently found to be bad.)

### **4.2 DEFAULT OPTION IPL**

The following steps describe a default option IPL. The default option IPL always loads the first copy of PASS system software. As one might expect, the hardware must be configured in a certain manner prior to starting either kind of IPL. The GPC POWER switch must be ON, the OUTPUT switch must be in NORMAL, and the MODE switch must be in HALT. In addition, the IPL SOURCE switch must be selected to the position for one of the MMUs (usually MMU1) and the BFC CRT DISPLAY switch must be in OFF. (If this latter switch is ON, the GPC will do a menu option IPL. This will be covered in detail in the next section.) After the switches are configured, the IPL process is started by pushing the IPL pb for that GPC. (Do not hold the IPL pb in longer than two seconds, since doing so might result in the IPL "hanging up" and not completing.)



The MODE tb for the GPC being IPL'ed will read "IPL", indicating that a small program called the bootstrap loader is being loaded in from the selected MMU. This program knows all the locations of all the software on the MMUs. (This way, the GPC hardware only needs to know one MMU address, that of the bootstrap loader. The addresses of the rest of the programs on the MMU change from flight to flight.) Once the bootstrap loader is loaded into the GPC's memory successfully, the GPC's MODE tb goes bp. The MODE switch is then taken to STBY, which tells the bootstrap loader program to load several other small programs. These include the self test program (STP), which checks the GPC's hardware; the GPC IPL program, which oversees the rest of the IPL process; and the system software loader (SSL), which loads the PASS system software.

Once the self test is complete, control is passed to the SSL which loads copy 1 of the PASS system software into the GPC. The MODE tb will now change to RUN, indicating that the IPL is complete and the GPC is ready to be taken to RUN. (This would also be a good time to take the IPL SOURCE switch back to OFF to free up the MMU for use by the other GPCs.) The MODE switch is then taken to RUN to start the PASS system software processing in this GPC. When this happens, the GPC will listen to the sync discretes and data buses to determine if there are any other PASS GPCs currently processing. If there are, it will try to join in a common set with them. If no common set is detected, the GPC will take control of DK buses 1 through 3 (displaying the GPC MEMORY page on the DPS displays) and form its own "common set". The OUTPUT tb will change to gray, indicating that the GPC is capable of commanding the flight critical data buses.

The final actions taken for a default IPL are done to ensure that the GPC successfully joined the common set (obviously not needed if this is the first GPC that is IPL'ed). A selected IDP/CRT MAJ FUNC switch is placed in PL, an unsupported major function, and the GPC is "hard assigned" to the display via the GPC/CRT key. The PL major function is selected since the newly IPL'ed GPC will not be able to drive a display with its MAJ FUNC switch in GNC or SM if any other GPC is processing GNC or SM applications software. Since any command using the GPC/CRT key is information that is exchanged between all GPCs in the common set (common set info), the newly IPL'ed GPC will "hear" the GPC/CRT key assignment and take command of the display. If this process succeeds, it is proof positive that the newly IPL'ed GPC is in the common set. The GPC/BUS STATUS display (SPEC 6) will also show information for this GPC rather than a blank column. Now that the GPC has been IPL'ed, it is ready to have the appropriate applications software loaded.

### 4.3 MENU OPTION IPL (PASS)

The menu option IPL has all the same basic processes as the default option IPL, with the additional ability to choose which copy of PASS or BFS software will be ultimately loaded into the GPC. The menu option IPL process begins the same way as the default option, except for one switch configuration change. The BFC CRT DISPLAY switch is placed in ON for a menu option IPL. The combination of this switch with the IPL SOURCE switch being selected to one of the MMUs, not only tells any other GPCs to leave that MMU alone (like in the default option IPL), but also tells the BFS not to command the display specified by the position of the BFC CRT SELECT switch. In addition, the BFC CRT DISPLAY in ON tells the IPL'ing GPC to command the display specified by the position of the BFC CRT SELECT switch. The selected display will have a Big-X and a POLL FAIL displayed once the switches are correctly positioned. One further step is needed in the menu option IPL process at this point – an IDP LOAD must be performed for the selected display. This is accomplished as a part of the IPL process. To have the IDP request its load from the IPL'ing GPC, one needs to toggle the appropriate IDP LOAD switch on Panel O6. The DPS display for the IDP will then display the words "VM LOAD IN PROGRESS." (The IDP LOAD is necessary, even if it has already been loaded, because menu option IPL software expects to load an IDP and will not continue without doing so.) The IPL process steps are then the same as in the default option IPL until the GPC is taken to STBY.

Once the GPC is in STBY (and all the necessary software has finished processing, including the loading of the IDP), the selected display will show the GPC IPL MENU (1) page. There are normally two "fault" messages in queue - one to say that the display software was loaded and the second to say that the GPC IPL program was loaded. Pressing MSG RESET twice will clear them.

The GPC is now ready to load in the desired copy of PASS system software. This is selected via item 1, 3, or 5 (for PASS copies 1, 2, or 3, respectively) on the GPC IPL MENU (1) display. After the item is entered, a Big-X and POLL FAIL will appear on the display and the SSL will load the desired copy of PASS system software into the GPC.

When the software is successfully loaded, the GPC's MODE talkback will change to RUN. At this point, it is a good idea to place the BFC CRT DISPLAY switch back in OFF to free up the display for use by the other GPCs. As before, the IPL SOURCE switch should be taken to OFF to free the MMU up for use by the other GPCs. The GPC's MODE switch is taken to RUN, its OUTPUT talkback goes gray, and the GPC comes up in OPS 0 and joins the common set. As with a default option IPL, the GPC is hard assigned to a display to verify it has joined the common set.

```

          GPC IPL MENU (1) 1 122/L3: 17: 53
PASS1 1 PASS5 9          Q02/L1: 17: 53
BFS 2 BFS5 10 STP/PURGE CYC CNT 0
PASS2 3 PASS6 11 BRRDR/MSG COD# 132
BFS2 4 BFS6 12 BRRDR/MSG COD# CNT 2
PASS3 5 PASS7 13 TOTAL BRR/MSG CNT 2
BFS3 6 BFS7 14 MSGS STILL IN LIST 2
PASS4 7 PASS8 15
BFS4 8 BFS8 16
          MCDS BIT#
          MOD#
          BSR1 8200
          BSR2 8000
          SWSW 0000

17 DEU FORMAT LOAD -

START GPC SELF T#ST 18
STOP GPC SELF T#ST 19
START DEU SELF T#ST 20
STOP STP ON ERROR 24
CONTINUE STP ON BRR 25
INHIBIT CKPT LOAD 26

          27 OPTION -
          START 28
          STOP 29
    
```

TD6D104D1. CRT# 1

**Figure 4-1. GPC IPL MENU display**

#### 4.4 BFS IPL

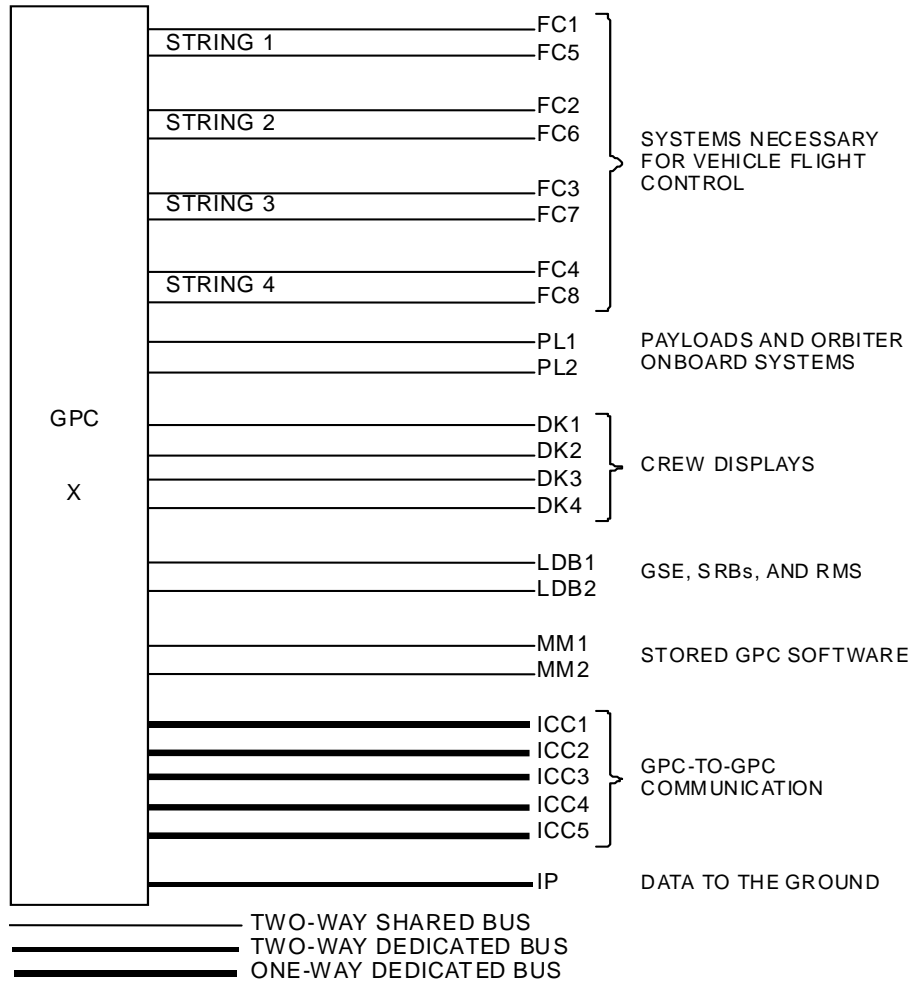
IPL'ing the BFS is basically the same as a PASS menu option IPL; however, once the GPC IPL MENU (1) display appears, a choice can be made to load a copy of BFS software. This will begin a process of loading the BFS software into the GPC. Since the SSL that is in the GPC at this point is only able to load PASS system software, the GPC IPL program will load the backup system loader (BSL). The BSL is fully analogous to the PASS SSL and knows how to load the BFS software. Once the BSL is loaded, the title of the display will change to reflect that and a "fault" message will be annunciated as well. The process of loading the BFS software into the GPC is started by clearing this message, which is done by pressing the MSG RESET key. When this is complete, another message will be annunciated to tell you that the load of the BFS software is complete. When the MSG RESET is pressed to clear this message, one of two things will happen. If the IPL SOURCE switch is already in OFF, the GPC will start the BFS software processing. If the IPL SOURCE switch is still selected to one of the two MMUs, a message will be annunciated to tell you to place the IPL SOURCE switch in OFF, which will start the BFS software processing. It is important to remember that a BFS IPL loads all of the BFS software. Unlike PASS, there are no further applications software loads to be done, so the BFS is now ready to respond "instantly" to an OPS transition request and process applications software.

## 5.0 DATA BUSES

Serial digital data are exchanged between the GPCs and other orbiter electronic devices via data buses (see Figure 5-1). A data bus physically consists of a pair of insulated wires that are twisted together and then electrically shielded to reduce the chances of electromagnetic interference affecting the signal. The electronic devices connected to a data bus are generically referred to as bus terminal units (BTUs).

There are seven groups of data buses used in DPS operations:

- a. **Flight critical (FC)** buses (eight) communicate with devices critical to flying the orbiter, both on orbit and in the atmosphere and providing flight instrument data to the Integrated Display Processors (IDPs).
- b. **Payload (PL)** buses (two) communicate (via MDMs) with devices to control payloads and various orbiter onboard systems (e. g., comm system antennas, fuel cells, payload dedicated flex MDMs, etc).
- c. **Mass memory (MM)** buses (two) link the GPCs to the mass memory units (MMUs); the magnetic tape drives that store the software for the GPCs.
- d. **Intercomputer communication (ICC)**, occasionally referred to as IC buses (five), allow GPC to GPC communication to support the common set and redundant set synchronization schemes.
- e. **Display/keyboard (DK)** buses (four) communicate with IDPs, which in turn drive the DPS displays on the Multifunction Display Units (MDUs). PASS can support up to three active DK buses at any given time.
- f. **Instrumentation/pulse code modulation master unit (IP)** buses (five) communicate between the pulse code modulation master units (PCMMUs) and the GPCs. The PCMMU routes the downlisted data streams from the GPCs into the orbiter communications systems for transmission to the ground and routes data from various orbiter onboard systems to the GPCs.
- g. **Launch (LDBs or LBs) data** buses (two) communicate with the ground support equipment (GSE) and the launch multiplexer/demultiplexers (MDMs) prelaunch and with the solid rocket booster (SRB) MDMs during powered flight. On orbit, one launch data bus is used to control the remote manipulator system (RMS), which is the robotic arm carried in the payload bay.



td601 a 010.cnv

**Figure 5-1. GPC data buses**

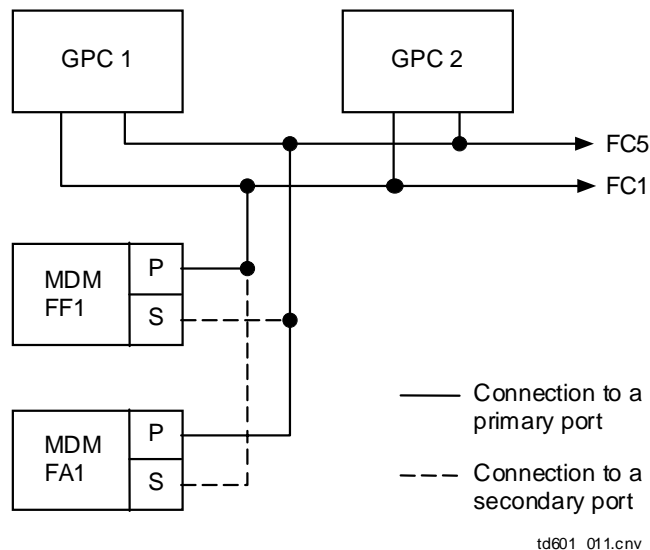
There is another fundamental concept relating to data buses that should be discussed at this point. A given data bus may be commanded by one and only one GPC at any given time. Since the redundant set may consist of multiple GPCs (all of which must receive data from those buses commanded by any GPC in the redundant set in order for the GPCs in the redundant set to have the same data), the commander/listener system was developed. In this system, the GPC designated by the Nominal Bus Assignment Table (NBAT) for the current OPS for a given bus is the commander and is the only GPC to transmit on that bus. However, all the GPCs in the redundant set are listeners on the bus. The GPCs that are not the bus commander for a given data bus wait while the bus commander transmits the commands to the MDM on that bus and then all listen for the MDM's response. The listener GPCs do check to see if the bus commander actually transmits over the bus, but other than that do nothing else while the bus commander does its transmitting. This keeps the timing among the redundant set GPCs equal, which is necessary for them to remain in redundant set operations.

## 6.0 PORT MODING

Each of the FF, FA, and PL MDMs has two multiplexer interface adapters (MIAs), more commonly referred to as ports. Each port is connected to a separate data bus, so there are two pathways (a primary and a secondary) by which the GPCs can communicate with a given MDM.

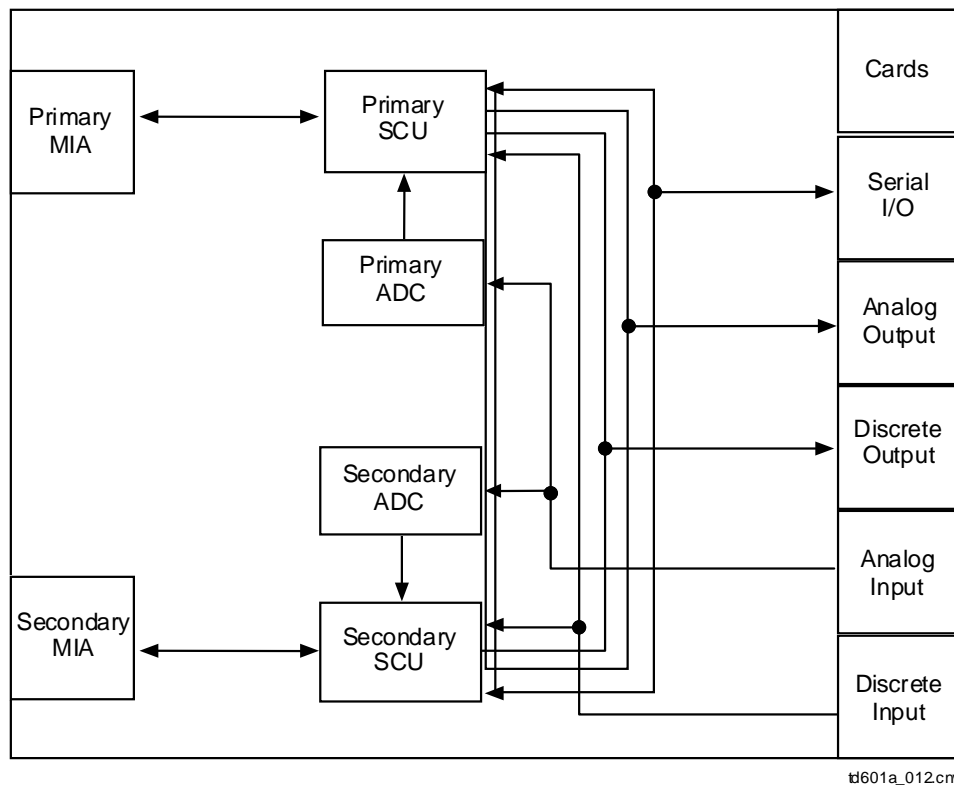
As an example, consider the FC MDMs on a given string. Each of the two FC data buses on that string is connected to both the FF and the FA MDM. (The design is the same for the two PL data buses and the PL MDMs.) However, each bus is used to communicate with only one MDM of the pair at a time. **Port moding** is a software method that controls which bus is used to communicate with which MDM (i. e., which port of each MDM will be active).

Initially, all strings and the payload bus pair are in the primary mode. Figure 6-1 represents the connections of the two flight critical buses to the two FC MDMs for string 1. Notice that the FC1 bus goes to the primary port of the FF1 MDM and the secondary port of the FA1 MDM. Therefore, the FC5 bus goes to the secondary port of the FF1 MDM and the primary port of the FA1 MDM. The solid lines going to the MDMs are to represent the pathways used when primary ports are selected. The dashed lines represent the pathways used when the secondary ports are selected. It should be emphasized at this point that port moding involves no hardware reconfiguration, only a software change in the GPCs as to which bus will be used to communicate with which MDM of the pair.



**Figure 6-1. MDM port connections on a string**

The reason that this scheme was implemented in the first place is that MDMs can have failures. As can be seen in Figure 6-2, an MDM is internally redundant for much of its circuitry. Thus, if one "side" of an MDM fails, port moding the GPCs over to using the alternate data bus to talk to the MDMs on a string will access the other "side" of the affected MDM. This should recover full function. This port moding process is initiated by the crew via SPEC 1 (DPS UTILITY) using items 15 through 24. If the problem had been a failure of the primary MIA on MDM FF1, item 16 would result in a port mode of string 1. Since port moding always involves both FC data buses on a string, both MDMs will be port moded at the same time. Thus, the GPC would now communicate with the FF1 MDM over the FC5 data bus and with the FA1 MDM over the FC1 data bus. This change would occur very quickly and would include the equivalent of an automatic I/O RESET EXEC on the port moded string only. Since both "sides" (or ports) of FA1 work just fine and the secondary port of MDM FF1 works just fine, full communication would be restored. It is also important to remember that control of the other BTUs on this string is unaffected by the port mode process. (A port mode of a string will also disrupt the process of the BFS listening or "tracking" on the buses of that string. This will be covered more fully in the next section.)



**Figure 6-2. MDM internal redundancy**

## 7.0 BFS TRACKING

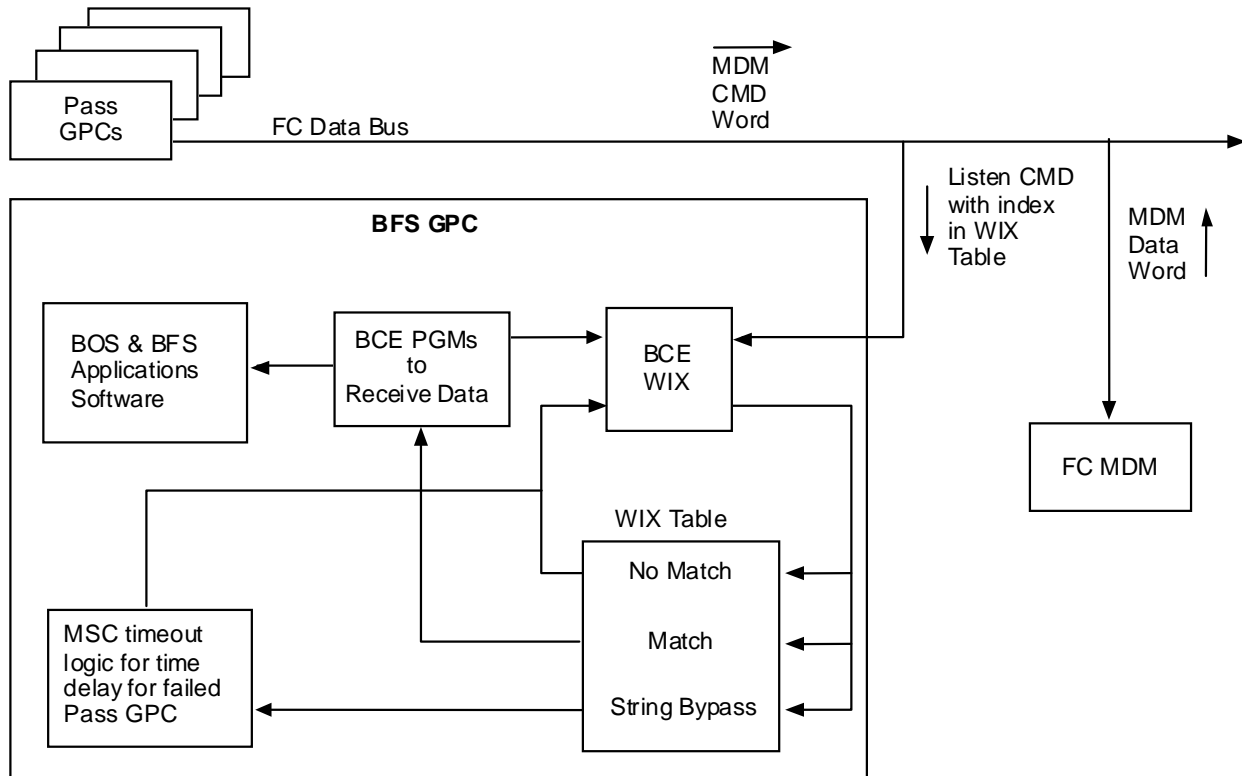
Since the Backup Flight System (BFS) GPC is never in a common set or a redundant set with the other GPCs, it does not exchange information like the PASS does. The BFS can never send PASS data, but it can receive data from the PASS. If the BFS is not engaged, it is also dependent upon the PASS to poll the MDMs, etc., in order to receive the data it needs. (Of course, if BFS is engaged, it will take control of the buses, itself, and poll the BTUs for its own data.) Since the non-engaged BFS must listen to the responses of the MDMs, etc. to PASS polling in order to get its data, it must somehow know when to listen. The process that allows BFS to accomplish this is called **tracking**.

The establishment of the tracking state is a three-step process. The first step is for the BFS to **sync** with the timing of the PASS outputs on the flight critical data buses. It does this by "listening" to PASS's transmissions in order to "hear" PASS poll, for example, port number 1 of the left Space Shuttle Main Engine (SSME) engine interface unit (EIU). Obviously, PASS must be processing an OPS that polls the EIUs. These are OPS 1, 3, and 6. Since the BFS "knows" that this poll occurs at a given point in the input/output profile, it knows when to listen for the next thing it needs.

The second step in the tracking process is to listen on FC buses 5 through 8 for data that PASS transmits solely for use by the BFS. This data contains such things as the orbiter's state vector and each PASS GPC's view of what BTUs' are responding to polling. The BFS compares this latter data between strings in order to determine if all the PASS GPCs are working correctly. If the data sent to the BFS by one of the PASS GPCs differs from that sent by the others, then the BFS knows that the PASS GPC in command of the bus it received the different data on is "bad", and the BFS will not try to track that string. In addition, it is necessary for BFS to be able to track at least two strings at a minimum. If the comparison of the data that PASS transmits for BFS's use is not a match for at least two strings, BFS cannot track PASS at all. This second step is referred to as the **track** step. Do not let the similarity of the name of this single step confuse you. It is only one step in the overall tracking process.

Once the BFS has decided which GPCs/strings to track, it starts to perform the third step in the process, **listening**. The listening process is performed continuously after that. The mechanism for BFS listening is a process called "Wait for Index" or WIX. WIX is a selective listen mode. By this, it is meant that the BFS listens to all the transactions occurring on the FC data buses and uses only what it needs. Prior to polling any given BTU for data, the PASS will transmit a WIX "listen command" on the same data bus. The BFS compares the index specified in this listen command to a table it keeps in its own software. If the index is in the table, the BFS needs data from this transaction and will listen to PASS command the BTU and also to the response from the BTU. If the index is not in the table, then the BFS will simply wait the correct amount of time for the transaction to complete and then listen for PASS to send the next listen command. For example, BFS wants the data returned by MDM FF1 from inertial measurement unit 1 (IMU1) when PASS polls it and will listen for the command to the BTU and its response when it sees a listen command for this. However, when PASS sends the listen command for microwave landing system unit 1 (MLS1), BFS knows it does not use this data and will simply wait for the next WIX listen command. The diagram in Figure 7-1 graphically summarizes this process.





td601a\_013.cnv

**Figure 7-1. The BFS listen process**

In summary, the three conditions that must be met in order for BFS to track PASS are: (1) BFS must be able to match timing sync with PASS, (2) BFS must be able to see the data PASS transmits specifically to it and that data must be identical on at least two strings, and (3) PASS must be processing GNC OPS 1, 3, or 6 so that it will be sending out the WIX listen commands that the BFS will need in order to know when to monitor for the data returned by the various BTUs. If all of these conditions are not met, the BFS cannot track PASS, and will "go standalone". If this happens in flight, the BFS cannot get the data it must have in order to be able to take over control of the vehicle in the event that the need to engage the BFS arises. The most important loss of data is from the navigation sensors, such as the IMUs. Without this data (for as little as 10 seconds during the powered flight portion of ascent or 45 seconds during atmospheric flight in the entry phase), the BFS is considered under the Flight Rules to be "no-go" (not fit to be engaged), due to the degraded condition of its state vector (mathematical representation of the orbiter's current location and velocity). Also, since tracking is so critical to the operation of the BFS, the BFS will reject an OPS transition request from OPS 0 to OPS 1 or OPS 3 if it cannot track PASS (assuming BFS is not engaged). Thus, if an attempt is made to PRO BFS into OPS 1 or OPS 3 when PASS cannot be tracked (such as PASS in OPS 2), the BFS will momentarily enter the requested OPS, determine that it cannot track, and revert to OPS 0.

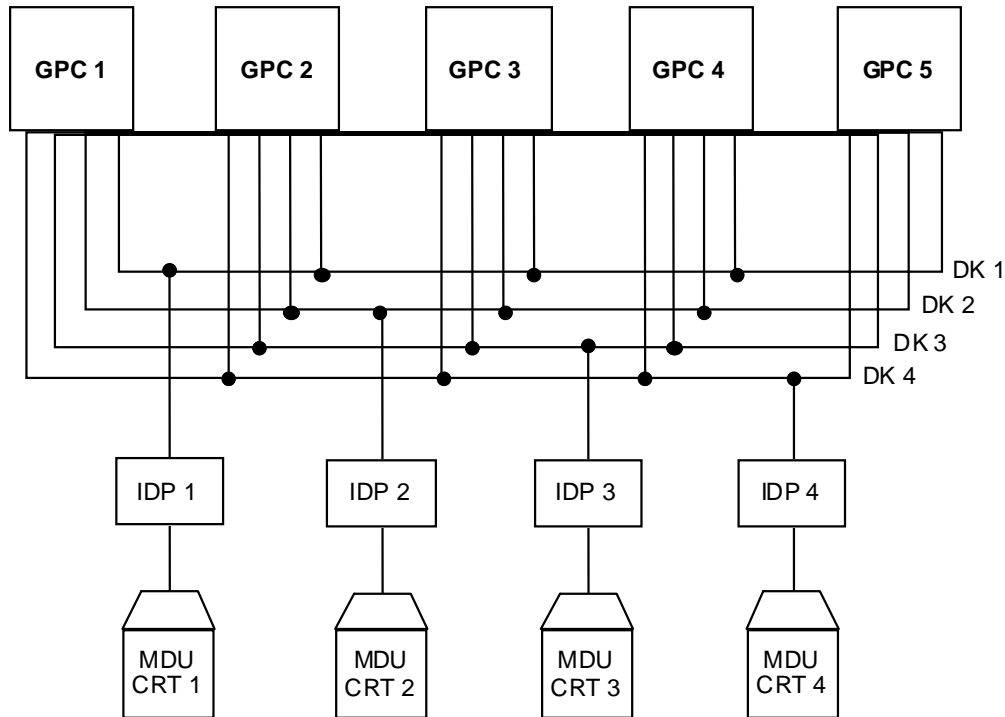
As was mentioned previously, port moding affects the BFS tracking process. The third stage of the BFS tracking process (the listening stage) involves the BFS listening for WIX indices from the PASS prior to each I/O transaction. Since port moding a string causes PASS to attempt each I/O transaction on the opposite bus of the affected string from the one it was using previously, the WIX indices are also moved to the opposite bus. When the BFS receives a different WIX index than it was expecting on two successive I/O cycles, it stops tracking (or "drops") that string. However, there is a method of recovery built into the BFS software. When the BFS receives an "I/O RESET EXEC" on a keyboard it is commanding, it does the three steps of the tracking process. One piece of information included in the data that PASS sends to BFS over FC buses 5-8 is the port mode "status" (primary or secondary) of each string. The BFS uses this information to know, in advance, on which flight critical bus to expect the sets of WIX indices to be transmitted. Since the BFS then receives only WIX indices that it expects, it will track properly again.

At this point, it is important to make a distinction between BFS tracking and the BFS "DK listen" process. The latter is the process whereby the BFS monitors crew keyboard entries made to DPS displays controlled by the PASS. The DK listen process goes on whenever the BFS is active; it has no conditions to meet.

## 8.0 DK BUS ASSIGNMENT

### 8.1 OVERVIEW

DK bus assignments, reassignments, and their impact on normal operations require special attention. The DK buses serve as a link between the GPCs and the IDPs for data displayed on the DPS displays and for keyboard inputs to the GPCs from the crew. DK bus assignments are often referred to as CRT assignments since the MEDS replaced the CRTs on the orbiter and the CRT MDUs are normally used for DPS displays. Figure 8-1 shows a representation of the DK buses.



td601a 014.cnv

**Figure 8-1. DK data buses**

The DK buses are similar to other data buses in that a GPC can listen to and/or command more than one bus at a time. Also, like the other data buses, a given DK bus (and therefore its associated IDP/CRT) can only be commanded by one GPC at a time. There is a complex set of software and hardware to ensure this. However, there is no such thing as a perfect system, and there are circumstances where two GPCs will attempt to command the same CRT at the same time. This condition is called **dual commanders** and can be recognized by seeing two different GPCs trying to display information simultaneously. This will be discussed in detail later in this section.

During normal operations, the IDP receives updates of currently displayed data from the GPC. In addition, the IDP accumulates crew keyboard entries and major function switch changes to send to the GPCs when polled for this input (twice a second). If the IDP determines that it has not been polled by a GPC for three consecutive seconds, it will generate a "POLL FAIL" message in the lower right-hand corner of the DPS display. Similarly, if the IDP has not received display update data from the GPC for three consecutive seconds, the IDP displays a big "X" from corner-to-corner of the DPS display. For example, when an IDP/CRT is deassigned (GPC software told not to communicate with the IDP), the IDP will detect the loss of data updates and GPC polling. This will cause it to display both the big "X" and the "POLL FAIL" message. (NOTE: Even though the IDP normally receives updates of the current time for display as part of the data from the GPC, a deassigned IDP will continue to display an updating time on the DPS display. This is because the IDP updates the time from its own internal clock and displays it.)

There are **four ways** to change DK bus assignments:

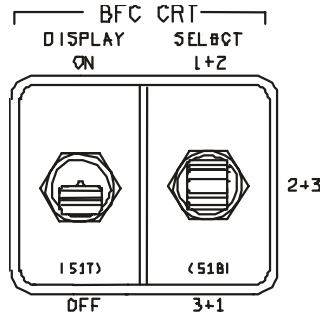
- a. BFC CRT DISP and SEL switches
- b. IDP/CRT MAJ FUNC switch
- c. GPC/CRT key
- d. OPS transitions/mode recalls

Each of these will be discussed in detail later in this section. In addition, the subsections will be presented in the order of priority that each topic has in the process of DK bus assignment.

## **8.2 BFC CRT DISPLAY AND SELECT SWITCHES**

Although the BFC CRT DISPLAY and SELECT switches (see Figure 8-2) have a unique function during a GPC IPL, their primary use is to allow easy assignment of a forward CRT to the BFS. Since this involves the use of switches to accomplish the CRT assignment, the mechanism is, per force, via hardware. The switches are routed through the backup flight controllers (BFCs) and result in a pair of discretes either being sent or not sent to all GPCs. Which discretes are on and which are off depends on the positions of the switches. When a PASS GPC in command of a forward CRT sees the discretes that correspond to the CRT that it is driving, the PASS GPC will stop driving the CRT. When the BFS sees the discretes corresponding to this CRT, it will take command of the CRT. (If the BFC CRT DISPLAY switch is off, both discretes are off and the BFS is not being told to drive a CRT via this method.) It should also be noted that the BFC CRT SELECT switch positions are labeled in an unusual manner. The positions are "1+2", "2+3", and "3+1". When the BFS is not engaged, it only takes command of the CRT specified by the first number of each of the switch positions. If the BFS is engaged, it will take command of **both** of the CRTs specified. (If the BFC CRT DISPLAY switch is off while the BFS is engaged, it will get CRTs 1 and 2, by default. This was implemented this way because, if BFS is engaged in ascent or entry, it is the **only** GPC active, and the crew must have interface to it.)

One interesting aspect of CRT assignment via these switches is that the CRT assignment within PASS (i. e., PASS's memory of which PASS GPC should have the CRT) is unaffected by giving BFS a CRT via the switches. Therefore, when the BFS CRT DISPLAY switch is moved to the off position, the same PASS GPC will pick it back up. If that same PASS GPC was to fail while the BFS had the CRT, then PASS would no longer control the display and a new PASS commander would have to be assigned.

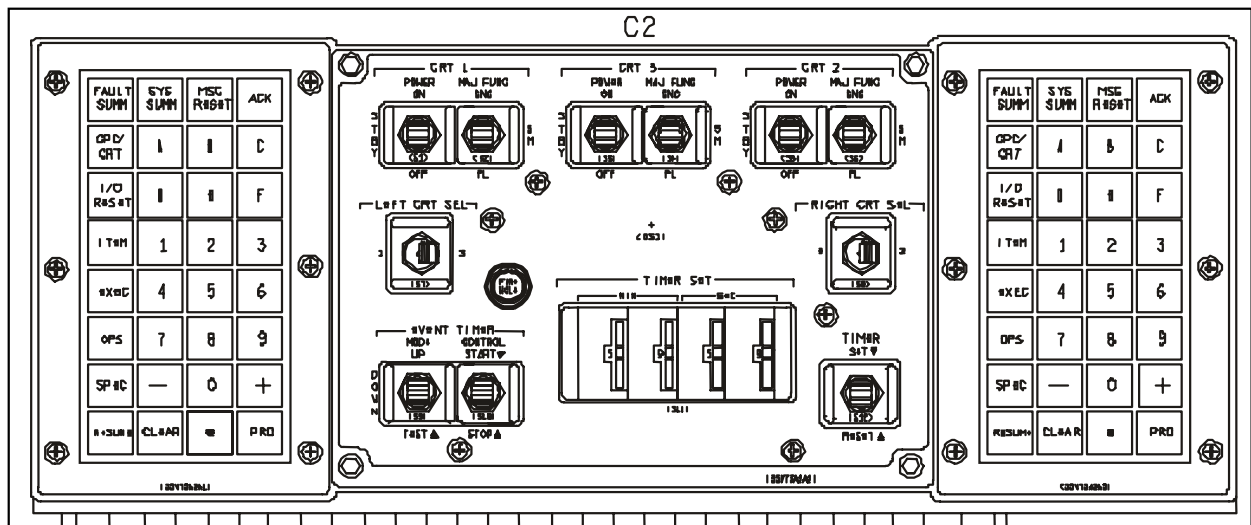


T06010103, PNL 3

Figure 8-2. BFC CRT switches on Panel C3

### 8.3 MAJ FUNC SWITCHES

The next highest priority in DK bus assignment is the IDP/CRT MAJ FUNC switch (see Figure 8-3.) While this method of CRT assignment is also implemented with switches, it is still predominately a software process. The MAJ FUNC switches are read by the IDPs, not the GPCs. When the GPC commanding a given IDP polls it for inputs (for such things as crew keyboard inputs), it also receives the current position of the MAJ FUNC switch. Thus, if a DPS display is not being polled by a GPC (i.e., displaying the "POLL FAIL" message), then moving the MAJ FUNC switch will have no effect. However, for the nominal case where the IDP is being polled, the commanding GPC (or entire redundant set, if GNC) will receive the current MAJ FUNC switch position.



T06010103, PNL 2

Figure 8-3. IDP/CRT controls and keyboards on Panel C2

If the commanding GPC is a PASS GPC, the first thing that it does upon receiving inputs from an IDP is to compare the current MAJ FUNC switch position with its own major function. If they are not a match, the GPC will check its common set information to determine if there is another GPC in the common set that is processing software in that major function. If there is, the GPC will stop commanding that IDP/CRT, update the common set information that it sends to the other PASS GPCs to reflect the new MAJ FUNC position, and stop listening to that DK bus as well. (If it is a GNC redundant set listener, it will just stop listening.) When the rest of the PASS GPCs receive this new common set information at the next common set synchronization, a GPC processing the appropriate major function will take command. If there is no GPC in the common set processing applications software in the major function specified by the new position of the switch (an unsupported major function), then the current commander retains control and displays the GPC MEMORY page.

If the MAJ FUNC switch of a given CRT was moved into GNC from some other major function, the GPC that formerly commanded the display to give it up and tell the common set that the MAJ FUNC switch is now in GNC. One of the redundant set GNC GPCs would then assume command of the display. Which GNC GPC takes control of the display depends on the Nominal Bus Assignment Tables (NBATs). In the on-orbit case, the DK bus portions (labeled CRT) of the two active NBATs (GNC OPS 2 and SM OPS 2) are as shown below:

	<u>MC 2</u>		<u>MC 4</u>
CRT	1 12 <u>1</u>	CRT	1 12 <u>4</u>
	2 13 <u>2</u>		2 13 <u>4</u>
	3 14 <u>0</u>		3 14 <u>0</u>
	4 15 <u>2</u>		4 15 <u>4</u>

The nominal DPS config on orbit has GPCs 1 and 2 being redundant set GNC GPCs. Thus, either could be the commander of a given IDP/CRT whose MAJ FUNC switch was placed in the GNC position. For example, the GNC OPS 2 NBAT specifies GPC2 as the commander of CRT 2 when the GNC major function is selected. Both GPC1 and GPC 2 would listen to the DK2 data bus. Since it is important to maintain interface with the GPCs, there is default logic in the PASS software that retains command of a display if an invalid GPC (or a zero) is specified for a given CRT in the NBAT. The rule is that the lowest numbered GPC in the redundant set will then take command of the display. No OPS/mode recall or OPS transition is required to invoke changes to the CRT portion of the NBAT for an active OPS; simply moving the MAJ FUNC switch is sufficient.

Since the NBAT can sometimes be misleading as to which GPC is actually commanding a CRT, the GPC/BUS STATUS display (SPEC 6) can be used to see the true bus assignments as shown in Figure 8-4. The asterisk for CRT 1 in the GPC 1 column indicates CRT 1 is being commanded by GPC 1. GPC 4 indicates a down arrow for CRT 1 which means that it is masking this bus (DK1) since it is not in the redundant set with GPC1. The absence of bus mask indications by GPCs 1 and 2 indicate that they are both listening to CRT 1 since they are both in the redundant set. Note that CRT3 is not currently assigned since the PASS common set can only command three IDP/CRTs at one time. Therefore all three active PASS GPCs show down arrows for CRT3 and no commander is indicated.

		GPC/BUS STATUS					2 213/17: 16: 37				
		1		2		3		4		5	
		MODE	RUN	RUN	HALT	RUN	HALT				
		OPS	GZ	GZ	0	SZ	0				
STRING	1	FF	*					↓			
		FA	*					↓			
	2	FF		*				↓			
		FA		*				↓			
	3	FF	*					↓			
		FA	*					↓			
	4	FF		*				↓			
		FA		*				↓			
	PL	1	↓	↓				*			
		2	↓	↓				*			
LAUNCH	1		↓	↓				*			
	2		↓	↓				↓			
CRT	1		*					↓			
	2			*				↓			
	3		↓	↓				↓			
	4		↓	↓				*			

TD6010903. CRT. 1

**Figure 8-4. GPC/BUS STATUS display (SPEC 6)**

The BFS software treats the MAJ FUNC switch differently from the PASS. Since the BFS has both GNC and SM functions in one GPC, it will drive the CRT with the appropriate display. The BFS has no PL software, however. So if a CRT with its MAJ FUNC switch in the PL position is assigned to the BFS, it will drive time fields on an otherwise blank display. As soon as the MAJ FUNC switch is moved to a position that BFS has software to support, it will drive the appropriate display. If the MAJ FUNC switch for a CRT already assigned to the BFS is moved from SM or GNC directly to PL, the BFS will just continue to display what it was displaying previously.

#### 8.4 GPC/CRT KEY

Use of the GPC/CRT key is third in priority as a method of assigning displays. This key is designed to initiate a sequence of keystrokes to allow the crew to assign a given CRT to a desired GPC. The keyboard entry sequence is "GPC/CRT XY EXEC", where "X" is the desired GPC and "Y" is the desired CRT (same order as the name of the key). This type of assignment is frequently referred to as "hard assigning" a CRT.

When a PASS GPC receives such a keyboard input, it relays it to the rest of the common set at the next common set synchronization point. Also, the GPC that initially receives the keyboard input from the IDP (this includes GNC "listener" GPCs, if the current major function is GNC) will analyze the request to see if the CRT reassignment is designed to give the CRT to some other GPC. If it is, the original commander will give up control of the CRT (much like the case of changing the MAJ FUNC switch discussed earlier). Since the content of the GPC/CRT key CRT assignment sequence is broadcast to the rest of the common set GPCs, the desired GPC takes control of the display. This process will take up to 0.16 seconds. If the original commander had been a GNC GPC and the desired commander is a redundant set GPC, the new commander would have heard the request directly as a listener GPC and acted upon it immediately.

A couple of effects of the design of this type of CRT assignment should be considered at this point. If the GPC/CRT key assignment command is made to reassign an IDP/CRT with its MAJ FUNC switch in a supported position (i. e., GNC or SM on orbit) and the desired GPC is processing a different major function, then the assignment will not happen the way it was expected. For example, if a display with its MAJ FUNC switch in GNC (i.e., commanded by a GNC GPC) is to be reassigned and the desired GPC commander is the SM GPC, then the following would happen: (1) the original GNC GPC commanding the CRT would hear the keyboard entry and would stop commanding it, (2) it would inform the common set of the CRT reassignment request, (3) the SM GPC would assume command of the CRT, (4) the SM GPC would poll the IDP and see that the MAJ FUNC switch is set to GNC and would stop commanding the CRT, (5) the SM GPC would inform the common set of the MAJ FUNC switch setting, and (6) the GNC GPC assigned in the current GNC NBAT would take command of the CRT. Since this all happens in less than a second, the DPS display would only appear to blink momentarily. This is because the MAJ FUNC switch assignment is higher priority than a GPC/CRT key assignment.

Alternatively, if the desired GPC is processing the same major function as that of the CRT's current commander (i. e., assigning a CRT from one redundant set GNC GPC to another), both the original CRT commander and the desired GPC hear the input simultaneously. Since the MAJ FUNC switch setting is the correct one for both GPCs, major function switch logic for CRT assignment does not apply. However, as discussed above, the commander for a CRT with its MAJ FUNC switch in the GNC position is supposed to be the NBAT-assigned GPC. How can these two opposing assignments be reconciled? The mechanism is a software function called the "NBAT override word". Each GPC maintains a record of whether or not a GPC/CRT key sequence has been entered by the crew for each CRT. If so, this record is used instead of the NBAT to assign the CRT. In effect, it overrides the NBAT assignment. This NBAT override word is cleared whenever the NBAT is invoked via OPS transitions and OPS/mode recalls.



Additionally, there is a third scenario to consider. What if the MAJ FUNC switch for the CRT being reassigned is in a position for a major function that is not currently active in any common set GPC (i. e., an unsupported major function)? Since major function switch logic is out of the picture for such a situation, there is no higher priority assignment method to override the GPC/CRT key assignment. Therefore, it would work just like it was intended. When trying to assign any OPS 0 GPC to a CRT, the MAJ FUNC switch is normally placed in PL, since PL is not normally used in flight and is therefore unsupported.

If a GPC/CRT key assignment is made with the BFS as the desired GPC, the former PASS commander will do the same things described above. It will release the CRT and tell the rest of the common set GPCs about the assignment. However, BFS does not listen to the ICC data buses to hear the information sent to the common set. It does know about the assignment directly, however. As discussed in Section 8, the BFS GPC listens to all DK data buses any time it is active—a process called DK listening. Thus, BFS will always hear any "hard assignments" made with the GPC/CRT key even if it is not the commander of the display. When it hears such an assignment, it will start polling the CRT. However, if the GPC/CRT key assignment sequence had been entered on a CRT being driven by the BFS (to assign a second BFS CRT, for instance) then bad things can happen. The PASS does not DK listen to BFS-commanded CRTs. Thus, the BFS would start commanding the CRT, but the PASS GPC currently commanding the CRT would not stop doing so. This would leave two GPCs trying to command the same CRT at the same time—dual commanders. Dual commanders are not a good situation to have. Not only is it difficult or impossible to read the data attempting to be displayed by either commanding GPC, the risk of causing a PASS redundant set fail-to-sync exists. (The mechanism for the possible fail-to-sync is data bus signal interference by the two commanding GPCs causing various redundant set GPCs to receive differing inputs from the IDP/CRT.)

In summary, the GPC/CRT key method of CRT assignment is third in the priority tree and is always subject to the BFC CRT DISPLAY and SELECT switches and MAJ FUNC switch logic. GPC/CRT key entries should always be made to a CRT commanded by a PASS GPC (even if the intent is to assign a second CRT to the BFS) to avoid CRT dual commanders.

## **8.5 OPS TRANSITIONS/MODE RECALLS**

The assignment of DK buses through an OPS/mode recall or an OPS transition is much the same as assignment of the strings and other buses through invocation of the NBAT. Therefore, whenever the CRT section of the current NBAT or the CRT section of the target NBAT is changed, an OPS/mode recall or an OPS transition will invoke the changes and the newly assigned GPC commanders will take over. It should be remembered, however, that this method is fourth in the priority tree and cannot override the BFC CRT DISPLAY and SELECT switch assignment or a MAJ FUNC switch logic assignment. As was mentioned earlier, it does clear the NBAT override word thus clearing any GPC/CRT hard assignments.

It should be noted that the NBAT cannot be used to cause PASS to assume control of a previously deassigned CRT and, likewise, cannot be used to cause PASS to give up a display it is currently commanding. A zero (or invalid GPC) in the CRT portion of the NBAT will result in control defaulting to the lowest numbered GPC running that software. This feature of the PASS system software was designed to prevent the possibility of CRT dual commanders resulting from an OPS transition.

## **8.6 SPECIAL CRT ASSIGNMENT RULES**

There are a few special circumstances to keep in mind, as well, for CRT assignments. The PASS is incapable of driving more than three CRTs at one time, no matter which method is used to assign the CRTs. If an attempt is made to assign the fourth CRT to any GPC in the common set, then the fault message ">3 DEU" will be issued and the request rejected. (DEU refers to the Display Electronics Units that were replaced by the MEDS.)

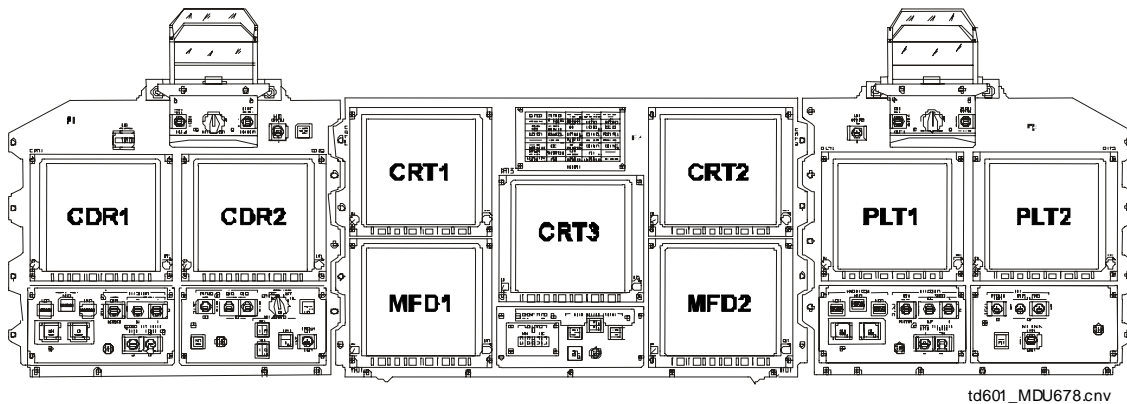
Special logic exists to allow any PASS GPC that is transitioning from STBY to RUN to assume control of CRTs 1, 2, and 3 in the event that it finds no existing common set or that the existing common set does not have control of any CRTs. This logic exists for the purposes of gaining interface to the first GPC IPL'ed pre-launch, gaining access to the first PASS GPC re-IPL'ed in the event of a BFS engage on ascent and subsequent PASS recovery, and regaining PASS CRT interface in the event that all CRT interface is lost. One example of the last scenario is to lose interface to all three assigned CRTs on orbit with the failure of the SM GPC when all assigned CRTs have their MAJ FUNC switches in the SM position (operationally avoided, as documented in the Flight Rules).

Another limitation related to CRT assignments is the availability of SPECs. Three SPECs can be supported in each Major Function. There is also a limit on system software SPECs (SPECs 0, 1, or 2) assigned to GPCs in a given major function, regardless of the positions of the MAJ FUNC switches for the assigned CRTs. Each system software SPEC can only be called once for any GPC. An example of this would be to move the MAJ FUNC switch of a CRT assigned to a GNC GPC into the PL position, (remember, the GNC GPC will retain control of the CRT), call up SPEC 0 (GPC MEMORY), move the MAJ FUNC switch back to GNC, and attempt to call GPC MEMORY again in the GNC MAJ FUNC switch position. The software would issue an "ILLEGAL ENTRY" message. These limitations can be avoided by always resuming SPECs prior to moving the MAJ FUNC switch.

## 9.0 MULTIFUNCTION ELECTRONIC DISPLAY SYSTEM

### 9.1 OVERVIEW

The Multifunction Electronic Display System (MEDS) is a recent upgrade to the orbiters that was designed to replace the original CRTs, mechanical gauges, and flight instruments. At the core of MEDS are the Integrated Display Processors (IDP). An IDP is an 80386-based computer that receives information from the GPCs over the FC and DK databuses as well as numerous analog voltage signals. The four IDPs use this information to support three different types of display formats (Flight Instrument, DPS, and Subsystem) on the eleven MEDS Multifunction Display Units (MDU). The MDU is a microprocessor controlled, six-inch square, flat-panel liquid crystal diode (LCD) display. The analog signals are converted for use by the IDPs via four MEDS Analog to Digital Converters (ADC). The nine forward-cockpit MDUs on Panels F6, F7, and F8, are shown in Figure 9-1.



**Figure 9-1. MDUs on Panels F6, F7, and F8**

The data received by the IDPs over the FC buses is used for Flight Instrument displays. The data received over the DK buses is used for DPS displays. The analog signals, after being converted by the MEDS Analog-to-digital converters (ADC), are used to display information on the Subsystem displays. In addition to displaying data, the IDPs can respond to the GPCs over the DK data buses to relay any keyboard inputs made to DPS displays by the crew.

## 9.2 MEDS DATABUS NETWORK

Figure 9-2, on page 9-3, shows the layout of the MEDS. Each IDP has an associated 1553 data bus to communicate with the various MDUs and ADCs. It should be noted that, unlike GPCs, IDPs operate independently and do not communicate with one another. With the exception of the CRT MDUs, each MDU and ADC is connected via two ports to two of the four IDPs. The dual connections on non-CRT MDUs allow for these displays to remain fully functional after a single IDP failure. The databus layout also ensures that no two IDP failures will take out all of the left or right MDUs.

The decision to limit the CRT MDUs to a single IDP connection was made to avoid any possible confusion over which IDP was driving a DPS display, especially when making keyboard entries. By dedicating the CRT MDUs as DPS displays, the knowledge of which IDP is controlling a DPS display is based on the location of the MDU. For instance, to make keyboard inputs to CRT 1 (controlled by IDP 1), the commander would place the LEFT IDP/CRT SEL (select) switch in the 1 (left) position – this is intuitive since the CRT 1 MDU is left-most DPS display. Non-CRT MDUs are fully capable of use as DPS displays, should the need arise; however, special care must be taken to ensure that keyboard entries are made to the correct IDP. The single-bus limitation is in the wiring of the data buses – the CRT MDUs actually have two fully functional ports.

The ADCs are connected and named such that the A-pair are connected to IDPs 1 and 2, while the B-pair are connected to IDPs 3 and 4. Each pair provides an identical set of information to the IDPs. In ADC failure cases, full capability can be recovered by forcing an MDU to swap IDPs to gain access to data provided by the other pair of ADCs. On the other hand, MDUs will normally automatically swap IDPs in the event that an IDP can no longer command the MDU. These operations will be discussed later.

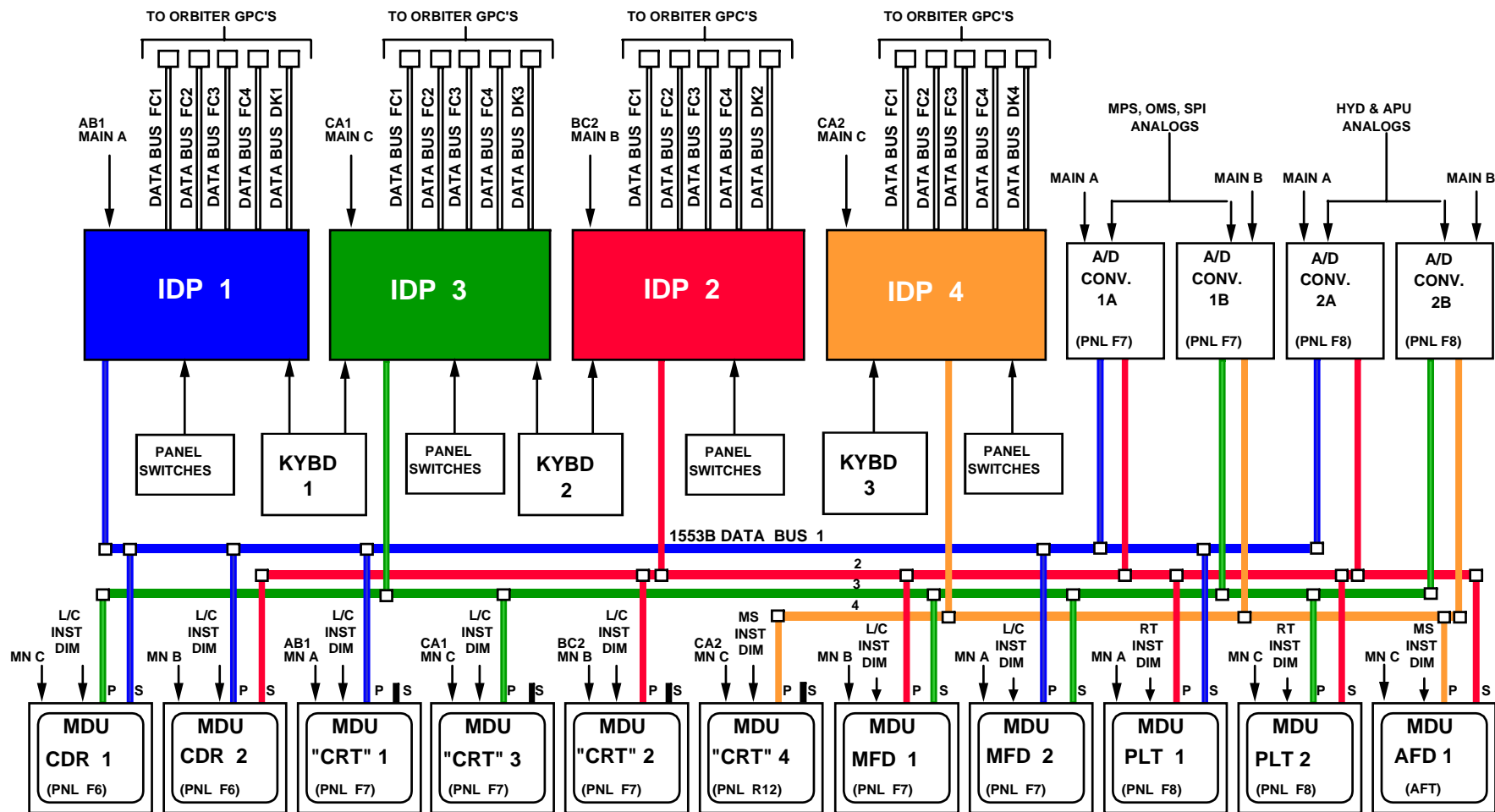
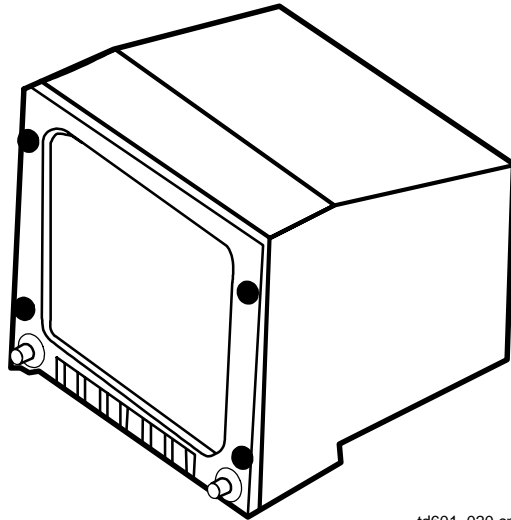


Figure 9-2. MEDS Data Bus Network

## 9.3 MEDS HARDWARE CONTROLS

### 9.3.1 MDU Controls

Each of the eleven MDUs has two controls on its faceplate or bezel, see Figure 9-3. To the lower left is the rotary power switch. On the lower right is the brightness control for the light bulb used to illuminate the LCD. Each MDU has six edgekeys along the bottom of its bezel to allow for navigation of the on-screen menu system.



td601\_020.cnv

**Figure 9-3. Multifunction Display Unit**

### 9.3.2 IDP/CRT Controls

Each IDP and associated CRT MDU have a pair of IDP/CRT POWER and IDP/CRT MAJ FUNC (major function) switches to control operations. The power switch is a simple OFF/ON switch; however, the switch provides power to the IDP and the CRT MDU. The major function switch has three positions (GNC, SM, and PL) and is used to determine which GPC will be commanding the DK data bus connected to the IDP, as discussed previously. The IDP/CRT switches for IDPs 1, 2, and 3 are located on Panel C2. The IDP/CRT switches for IDP 4 are located near the CRT4 MDU on Panel R11.

There are two keyboards on Panel C2 used for making inputs to IDP 1, 2, and 3 (usually for DPS displays). The LEFT and RIGHT IDP/CRT SEL (select) switches on Panel C2 are used to determine which of two IDPs will receive inputs from the associated keyboard. The left keyboard can make inputs to IDPs 1 and 3 while the right keyboard can make inputs to IDPs 2 and 3. IDP 4 has a keyboard of its own on Panel R11.

### **9.3.3 Other Controls**

The MEDS ADCs and non-CRT MDUs have circuit breakers on Panel R14 to provide power from the orbiter's main power buses. These circuit breakers are normally left in the closed position for all phases of flight.

There is a set of IDP LOAD switches on Panel O6 that provide a means of loading new DPS critical formats into the IDPs. These critical formats are used by the PASS GPCs as pre-loaded display layouts to minimize the amount of information that must be sent to the IDPs during dynamic phases of flight. These switches are used preflight when changes are made to the critical formats as part of an update to the PASS flight software and when performing menu-option IPLs of the GPCs.

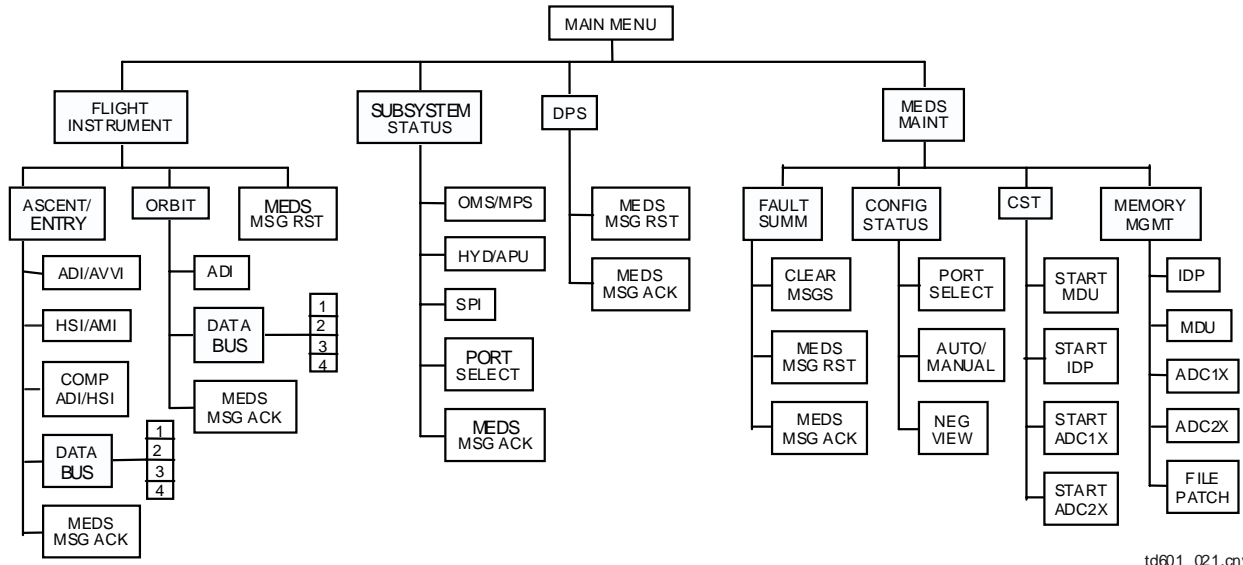
## **9.4 MEDS SOFTWARE**

### **9.4.1 Overview**

The MEDS IDPs and MDUs require software to be able to run. The software for these units is automatically loaded from electrically erasable programmable read only memory (EEPROM) into random access memory (RAM) when the units are powered up and the units should be available for use within approximately five seconds. The IDP can also retrieve its software from an internal hard disk in the event that its EEPROM fails. The ADC programming is stored in and runs directly from EEPROM and the unit should be available within one second of power-up. The ADC and MDU software is used to control the respective hardware and is largely transparent to the user. The IDP software controls the IDP hardware but also collects data from the data buses and ADCs, controls the user interface, and determines what will be shown to the user on the MDUs.

### **9.4.2 Edgekey Menu System**

The IDP contains the software which interfaces with the MDU to provide graphical and textual display information to the crew, receives the crew inputs via edgekeys, and controls the MEDS configuration. The edgekey menu system is used to control the MEDS system. The MEDS edgekey operations are organized by function and are displayed in the proper hierarchy depicted in Figure 9-4 (some options under the MEDS MAINT menu are not fully shown). In general, pressing an edgekey will call up an additional menu, select a MEDS configuration, select a particular MEDS display, or initialize internal tests of the MDUs, ADCs, and IDPs. Except at the Main Menu, the first edgekey (on the left) will be the up arrow which allows the user to go up to the next level. The main menu allows the user to select the flight instrument menu, subsystem status menu, the DPS display, and the MEDS maintenance display and its menu.



**Figure 9-4. MEDS Edgekey Menu Hierarchy**

Throughout the various menus, the user can select MEDS MSG ACK or MEDS MSG RESET. These edgekey operations are single stroke keys. The MEDS MSG ACK works for MEDS just like the keyboard ACK key works for the DPS. The first time that it is pressed, the MEDS fault messages will go static. Subsequent pressings of the edgekey will scroll through the remaining MEDS fault messages. The MEDS MSG RESET will clear the MEDS fault message line, just like the keyboard MSG RESET key works for the DPS displays.

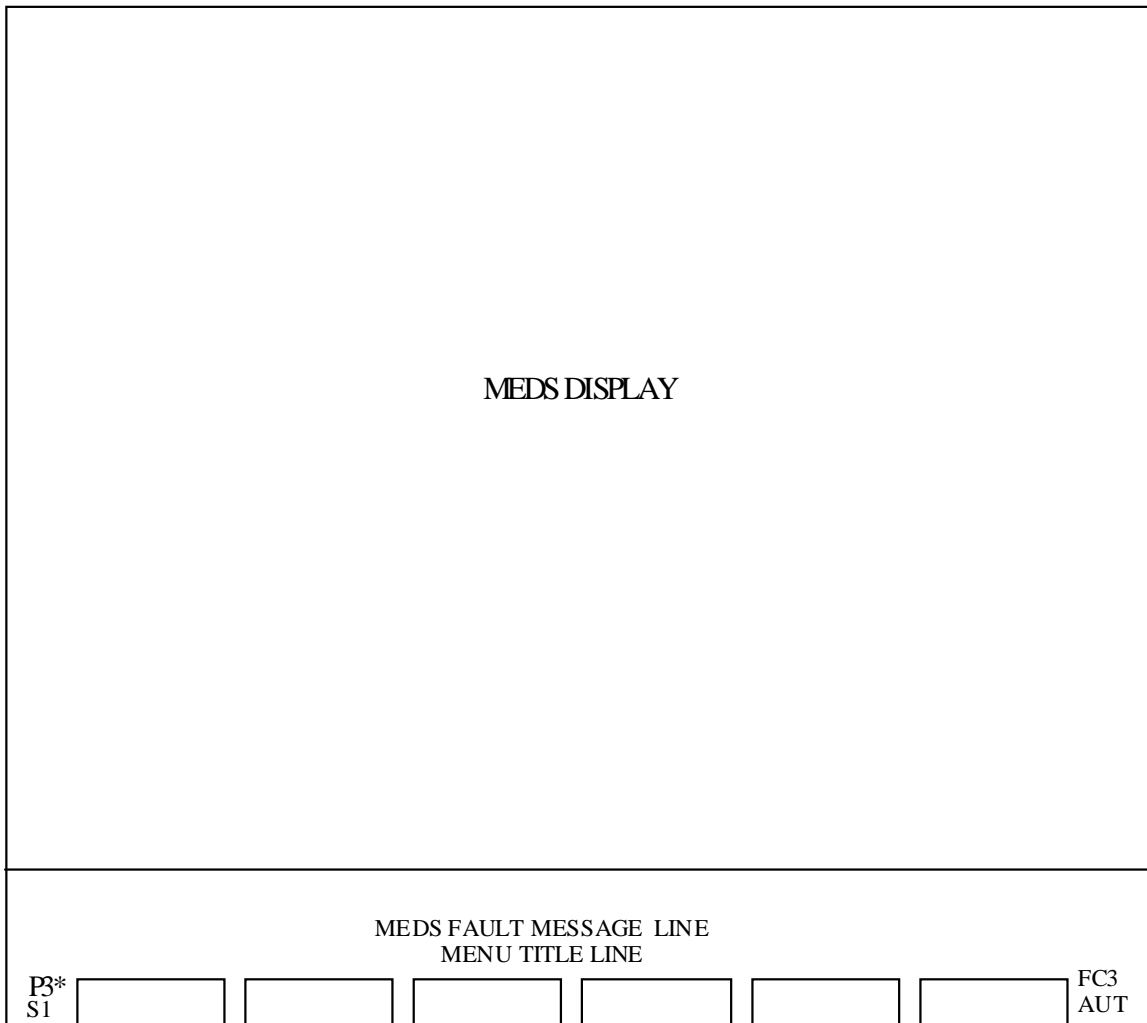
If an edgekey fails to the closed position or is held down for more than a few seconds, the edgekey will be disabled and the menu option associated with that edgekey will be unavailable. The MEDS will indicate a failed-closed edgekey by placing a red X over the associated menu option. Recovery of failed edgekeys can be attempted by power cycling the MDU.

### 9.4.3 MEDS Generic Screen Format

Each MDU screen will follow a generic screen format as seen in Figure 9-5. The top portion of the screen is denoted as the MEDS display format. The lower portion contains the MEDS internal configuration information. A horizontal cyan line will separate the two portions. The upper portion will be blank or display the selected MEDS display that will be discussed in a later section. At the bottom, the legends for the edgekeys will be displayed in the six boxes aligned with their respective edgekey. The color of the boxes and the labels will normally be cyan, unless they correspond to the current MEDS display, in which case they will be white. A blank edgekey legend means no option is available for that edgekey. If the edgekey is pressed, the IDP will ignore it. Above the edgekey boxes and legend, the menu title will be displayed. The MEDS fault message line is the line above the menu title. Any MEDS generated messages will be displayed in white on this line. Any GPC generated messages will be displayed on the DPS display in orange (above the cyan horizontal line).



MEDS configuration information will be displayed to the left and the right of the menu bars. The information to the left will indicate which MDU port is selected and which IDP is connected to each MDU port. "P" indicates primary port. "S" indicates the secondary port. The numbers next to the "P" and "S" indicates the IDP that is connected to that port of the MDU. An asterisk will indicate which port (and thus IDP) is selected to command the MDU. The information displayed to the right of the menu bars will indicate the port select configuration and flight critical data bus selected for that MDU. The flight critical data bus selected is the number displayed next to "FC". It should be noted that the FC bus is only applicable to flight instrument displays. The port reconfiguration mode will be displayed below the flight critical data bus information. "AUT" will be displayed when automatic port reconfiguration capability has been selected. "MAN" will be displayed when manual reconfiguration capability has been selected. Each MDU will retain its current configuration through a power cycle.



td\_601\_022.cnv

**Figure 9-5. MEDS Generic Screen Format**

### 9.4.4 MEDS Displays

The user may view all the flight instrument and subsystem status information via the various MEDS displays. The use and interpretation of the data displayed on the flight instruments and subsystem displays is beyond the scope of this workbook. Refer to the various disciplines associated with the data for additional information.

#### 9.4.4.1 Flight Instruments Displays

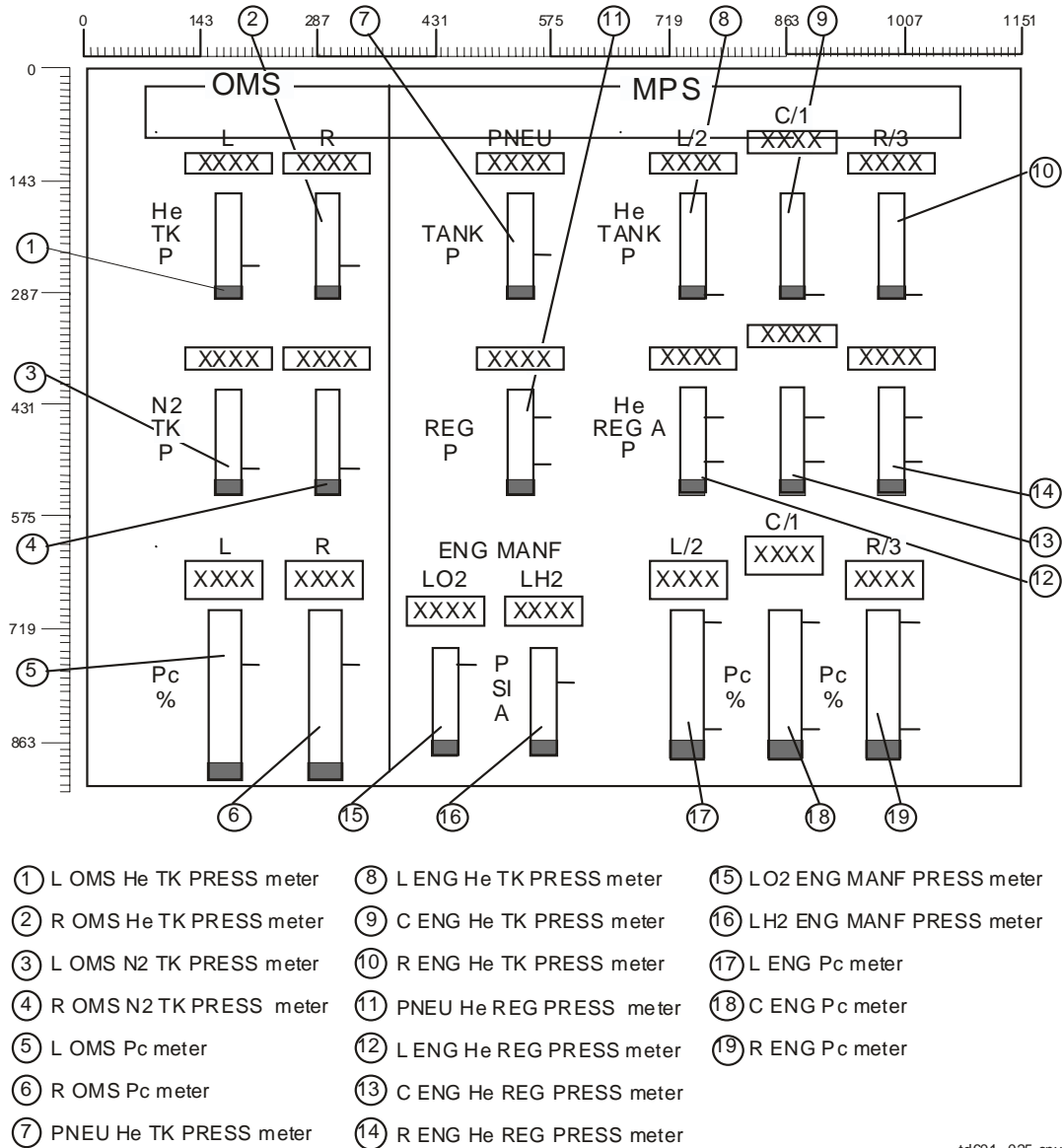
The ascent and entry flight instruments provide data about the orbiter's current attitude and position. This information is displayed on the Primary Flight Display (PFD). The PFD is tailored so it changes based on what data needs to be displayed for the current flight phase. Digital readouts are given for all the components on the PFD. The data shown on the flight instruments is provided to the IDP via the flight critical databus indicated in the lower right portion of the MEDS display and can be changed by selecting the DATA BUS edgekey. An example of the PFD is shown in Figure 9-6.



Figure 9-6. Example of Ascent/Entry PFD in MM 103

#### 9.4.4.2 Subsystem Status Displays

The three subsystem status displays will display the OMS, MPS, APU, and HYD meters and SPI in a digital and tape format. OMS and MPS data are displayed together, while APU and HYD data are together. The SPI is the third display. The displays consist of a digital readout and colored bar meters that indicate nominal (green/white) or off-nominal (yellow/red) conditions. Tick marks also indicate certain limits. In the event that the IDP controlling the display can no longer access an ADC, the data displayed on the associated Subsystem Status display would be blanked out. The user can attempt to restore data by pressing the PORT SELECT edgekey to force the MDU to swap ports to the alternate IDP, which accesses the alternate pair of ADCs (this may not recover data on the CDR2 and PLT1 MDUs, since both connected IDPs use the same pair of ADCs). The layout of the OMS/MPS display is shown in Figure 9-7.



td601\_025.cnv

**Figure 9-7. OMS/MPS Subsystem Status Display**

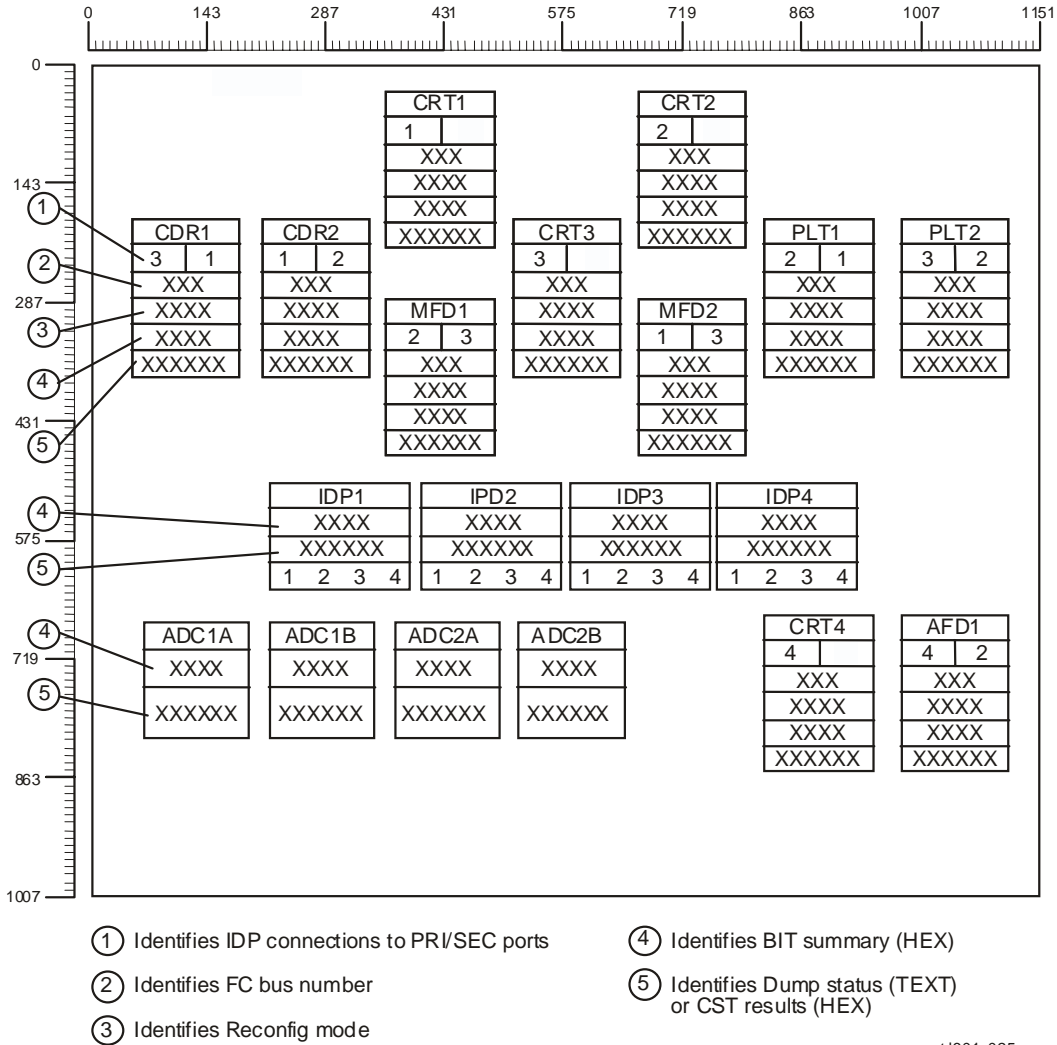
### 9.4.4.3 DPS Display

The IDP generates the DPS display using data it obtains over the DK bus. The DPS display is shown in green on black with the exception of special indicators. The Big-X, "POLL FAIL", and DPS fault messages are orange. Overbright characters are yellow. While DPS displays are normally located on the CRT MDUs dedicated to each IDP, the MEDS allows them to be placed on any MDU. It should be kept in mind that what the GPC considers "CRT X" is actually IDP X and, if there is any doubt, the IDP driving the display is indicated in the lower left corner of the MEDS status area on every display (see Figure 9-5). The LEFT and RIGHT IDP/CRT SEL switches must be selected to the correct IDP to make keyboard entries to a DPS display.

#### **9.4.4.4 MEDS Maintenance Display**

The MEDS maintenance display (see Figure 9-8) provides an overview of the current configuration of the MEDS system as viewed by the commanding IDP of the MDU. The MEDS maintenance display will show all of the MEDS LRUs as boxes. The forward MDUs are arranged on the display the same way they are arranged in the forward flight deck. Displayed below them are the four IDPs. The ADCs and the aft MDUs are displayed graphically below the IDPs. Each of the LRUs commanded by the current IDP will have a blue background for its box. Each LRU box provides information about its configuration, health, and/or Comprehensive Self Test (CST) results. Refer to the DPS H/W S/W 21002 workbook or the Shuttle Crew Operations Manual for additional information about the status indications and menu options on the MEDS maintenance display.

While most of the options under the MEDS maintenance display are used primarily for pre-flight configuration and testing, one option worth mentioning is the MEDS Fault Summary display. This display is the MEDS equivalent of the GPC Fault Summary. The display will list the past fifteen MEDS fault messages detected by the IDP controlling the display. Each IDP maintains a separate fault summary. IDPs can announce a variety of fault messages to inform the crew or the ground of errors. The IDP will generate a MEDS fault message for the categories of errors that warrant crew notification. The other categories will cause the IDP to notify the ground, archive the error entry, or inform the crew by a method other than a MEDS fault message. MEDS fault messages are normally displayed in the MEDS status area below the IDP's DPS display. If the IDP is not actively driving a DPS display, one of its other MDUs will be used for fault messages.



**Figure 9-8. MEDS Maintenance Display**

## 9.5 OPERATIONS

### 9.5.1 MDU Configuration

Normally, the MDUs will be configured to use their primary ports and will be in automatic port reconfiguration mode. CRT MDUs will display the DPS (“CRT”) display. In general, the CDR and PLT MDUs will display the flight instruments, while the MFD1 and MFD2 MDUs will be used for the subsystem displays. This configuration arranges the information displayed similarly to the older, non-MEDS cockpit.

## 10.0 DOWNLIST AND UPLINK

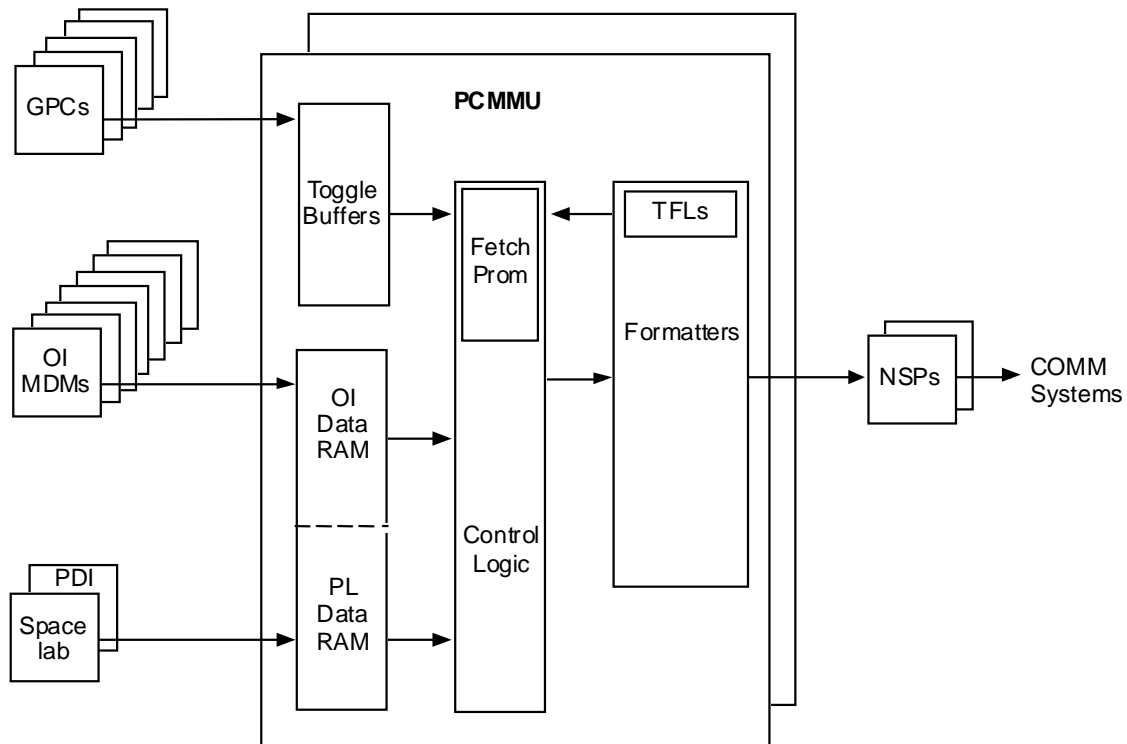
### 10.1 PULSE CODE MODULATION MASTER UNIT

Many of the DPS procedures related to post insertion, orbit operations, and deorbit prep involve loading Telemetry Format Loads (TFLs) into the Pulse Code Modulation Master Unit (PCMMU). The PCMMU is the device that routes the orbiter data to the OPS recorders and to the communications systems for downlink to the MCC. The TFLs are its programming. The TFLs will be covered in greater detail in the next subsection.

There are three general types of downlinked telemetry data:

- a. GPC downlist
- b. Orbiter systems data gathered by the Operational Instrumentation (OI) system
- c. Mission specific payload data

Their interfaces to the PCMMU can be seen in Figure 10-1.



td601a 026.cnv

Figure 10-1. Downlink flow through the PCMMU

The GPC downlist data is provided by the operating GPCs and is stored in special memory areas in the PCMMU called "toggle buffers". There are five toggle buffers and each has two independent sections so that the PCMMU can read one side of the buffer while it is receiving GPC downlist data on the other side. The functions of the two sides are then swapped, or toggled (thus, the name). This allows for a more rapid rate of downlist from the GPCs. The GPCs (normally only the SM and BFS GPCs) can also poll the PCMMU for data that it has accumulated from other sources (i.e., the OI system and payloads). This is the usual method for the SM GPC and the BFS GPC to obtain data to monitor the orbiter mechanical, electrical power, life support systems, and the payloads.

The PCMMU cyclically polls the OI MDMs to obtain data from the orbiter systems and stores this data in its memory. It also polls the Payload Data Interleaver (PDI) for payload data.

## **10.2 TELEMETRY FORMAT LOADS**

The TFL is the programming that tells the PCMMU which OI data, PL data, and GPC data to combine into either a high data rate (HDR) or a low data rate (LDR) output stream to be sent to the Network Signal Processors (NSPs). The active NSP will in turn route this data on to the orbiter's communications systems for actual downlink to the ground. The HDR stream is 128 kilobits/sec, and the LDR stream is 64 kilobits/sec.

The PCMMU has memory locations to store three TFLs. The first is a "hard" TFL, in that it is stored in non-volatile PROM memory and cannot be lost by powering off the PCMMU. The other two TFLs are user-loadable in real time. Thus, they have to be in RAM memory, and therefore are sometimes referred to as "soft" TFLs. The hard TFL is for a HDR stream. It is routinely used as the active HDR TFL for the ascent phase of flight. One of the two loadable TFLs is also for a HDR stream. The other is for the LDR stream. Since there is only one TFL memory available for the LDR stream, whatever TFL is loaded in that slot is always used to generate the LDR stream. A new soft HDR TFL (and also the LDR TFL) is loaded from time to time during a normal flight, depending on the varying needs for data to MCC during the different parts of a mission. SM SPEC 62 is used to load the soft TFLs. The procedure for selecting and loading TFLs is in the Orbit OPS Checklist, Flight Supplement.

## **10.3 GPC DOWNLIST**

Downlist (as opposed to downlink) is the data assembled by the GPCs' system software and sent to the PCMMU. In general, each active GPC sends its downlist to the PCMMU over its own dedicated data bus, see Figure 10-2. The exception is the redundant set GPCs. Since GNC GPC's are truly redundant, the downlist from each of them would be identical. Thus, only one copy of the GNC downlist is needed. Which GPC sends its copy is specified by ITEM 44 on SPEC 0, GPC MEMORY. This is usually the lowest numbered GNC GPC, since the software will automatically select that GPC at all GNC OPS transitions. Even in the event of a failure of the downlist GPC (so long as there were at least three GNC GPCs prior to the failure), the software will automatically select the lowest numbered surviving GNC GPC as the one to downlist.

Inside the PCMMU, the downlist is routed to the toggle buffers (discussed earlier). The number of a toggle buffer does not necessarily correspond to the number of the GPC

commanding it. The desired toggle buffer address is part of each GPC's downlist. A convention for which type of downlist goes to which toggle buffer was established to make coding TFLs easier. The downlist from the GNC "downlister" (the GPC specified to be the only GNC GPC to actually downlist) is sent to toggle buffer 1, the SM GPC's downlist goes to toggle buffer 2, and the BFS GPC's downlist goes to toggle buffer 5. (Toggle buffer 3 is never used, and toggle buffer 4 has certain special uses.)

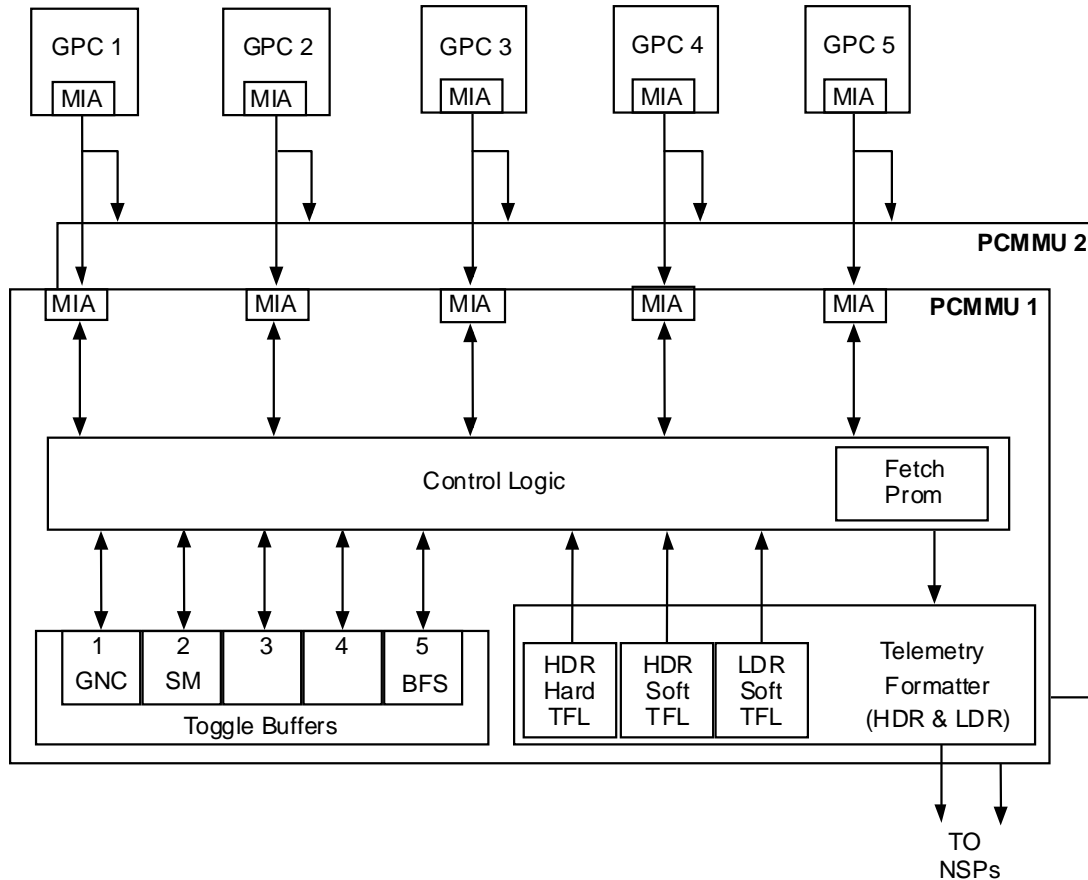
The types of data included in the various GPC downlists include data on the DPS systems themselves, crew station hardware status (such as switch positions), and major function-dependent orbiter systems or payloads data. BFS provides much the same type of data in its single downlist data stream.

As an example of these interfaces, let us assume that the GNC downlist has ceased to be received at MCC, but the rest of the downlink is present. The problem could be either at the GNC downlist GPC or at the active PCMMU. Since the crew would not even know that the problem exists until the ground tells them, the MCC would have to ask for recovery actions. The first would be to try one of the other GNC GPCs as the downlister, since this is only an item on a SPEC (and therefore the easiest thing to try). If that did not help (i. e., the problem was not at the downlisting GNC GPC), the next thing to try would be to swap PCMMUs. This is done with the PCMMU POWER switch on PNL C3. Moving the switch to the "2" position would take power away from PCMMU1 and power up PCMMU2. Thus, the soft TFLs that had been loaded in PCMMU1's RAM memory slots would be lost. PCMMU2 would automatically initialize itself to the hard TFL. This should recover GNC downlist. In order to select the right mix of data for this point in the flight, the crew would then be asked to reload the soft TFLs that were lost switching PCMMUs.

#### 10.4 VARIABLE PARAMETERS

Most of the data to be included in the downlist is specified preflight, in fact, months in advance. However, since it is not possible to be 100% accurate in predicting what data parameters will be desired by MCC during flight, a capability was developed to provide real time changes in the downlist from the GPCs. There are ten "slots" in the downlist from each major function (or OPS 000 GPC), called ***variable parameters*** that can be changed from the preflight assigned parameters. The desired parameters from the GPCs' memories can be specified by the crew on SPEC 1, DPS UTILITY. However, since they are really for the MCC anyway, the MCC specifies which parameters are to go in the slots via uplink commands. Any changes to the variable parameters take effect immediately and will remain (even across OPS transitions) until changed again by MCC or the crew.





td601a\_027.cnv

**Figure 10-2. GPC downlist path through the PCMMU**

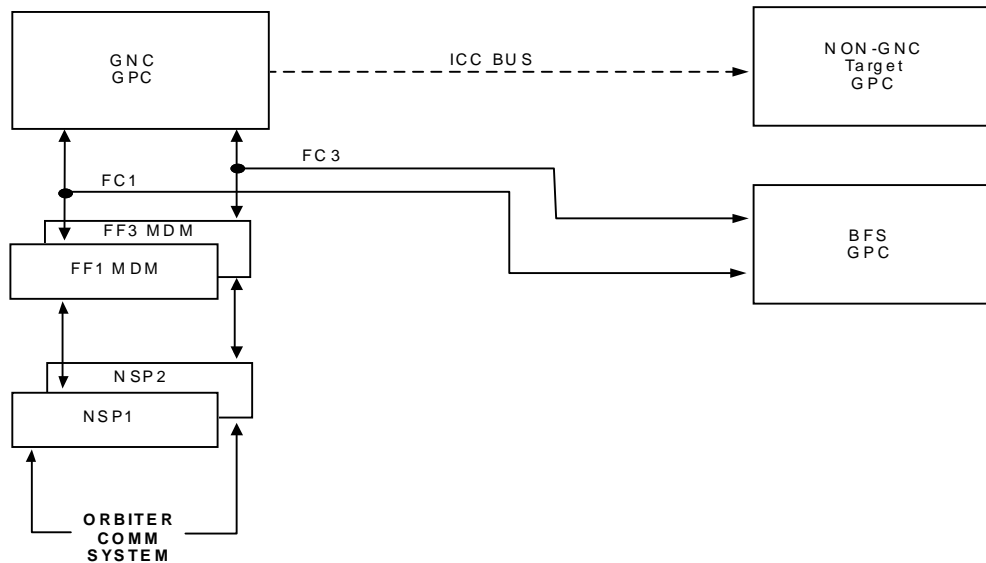
## 10.5 UPLINK

An uplink is a ground-generated command that is transmitted to the orbiter in flight (via radio signal) for processing and execution by onboard software. The term "uplink" can also be used as a verb to refer to the *process* of sending such a command. When a command is uplinked, it is routed through the orbiter's communications systems to the active NSP, and from there to a GNC GPC. The GNC GPCs check the uplink to determine which common set GPC is the intended target and send the command on to it via the ICC data buses. If the uplink was intended for the GNC GPCs, then they do not send it on, they just process it. If the uplink was intended for the BFS (pre-engaged), then the BFS heard it directly since it was tracking the flight criticals. In this case, the GNC GPCs do nothing further with the uplink. In the event of an uplink to the BFS once it is engaged, the BFS will be the GPC polling the NSP and will receive the uplink directly. Typical uplink commands include Inertial Measurement Unit (IMU) drift and accel compensations, state vectors, commands to payloads, and Table Maintenance Block Updates (TMBUs).

There are two basic types of uplinks. The first type, the **op code** method, is by far the more frequently used method of uplink. It causes the target GPC(s) to execute instructions already in the GPC's memory. The instructions to be activated are referenced by the dedicated op code (i. e., a number pointing to software subroutine to use) specified in the uplink (thus, the name). In addition, the uplink contains any data that will be needed by the instructions specified by the op code. A good example of an op code uplink is a TMBU, which is used to change certain SM fault detection limits, etc.

The second type of uplink, the **IDP equivalent**, is fundamentally different from the op code method. (In reality, the IDP equivalent is implemented via its own dedicated op code, but its uses are fundamentally different from other uplinks. So, it is considered as though it was a separate type.) The instructions it references are those that accept keyboard inputs. Thus, an IDP equivalent UL is *equivalent* to a keyboard entry made to an IDP/CRT. This gives the MCC the capability of "making" any keyboard entry the crew could. The target GPC(s) is "faked out" into believing that the entry came from a crew keyboard. (Perhaps the most unusual aspect of this capability is that the IDP that the target GPC can be made to believe was the source of the input can even be an IDP that is powered off.) Thus, the MCC *could* perform most keyboard entries that the crew does. However, that would not be as efficient, or as safe, as the crew doing so. In fact, the MCC ground computers inhibit certain keystrokes from uplink by IDP equivalents, due to safety concerns. The most frequent use of IDP equivalents is by the KSC Launch Processing System (LPS) prior to shuttle launches (e. g., the "OPS 101 PRO" to PASS). An example of an IDP equivalent uplink *sent by MCC* might be to clear a "trapped SPEC" from a power failed IDP by uplinking a RESUME key input.

During either type of uplink, a flashing "UL" will appear on the CRT(s) assigned to the target GPC(s), next to the GPC driver number. A diagram of the flow of an uplink through the DPS is shown in Figure 10-3.



td601a\_028.cnv

Figure 10-3. Uplink flow through the DPS

## **11.0 CHECKPOINTS**

### **11.1 CHECKPOINT INITIATE**

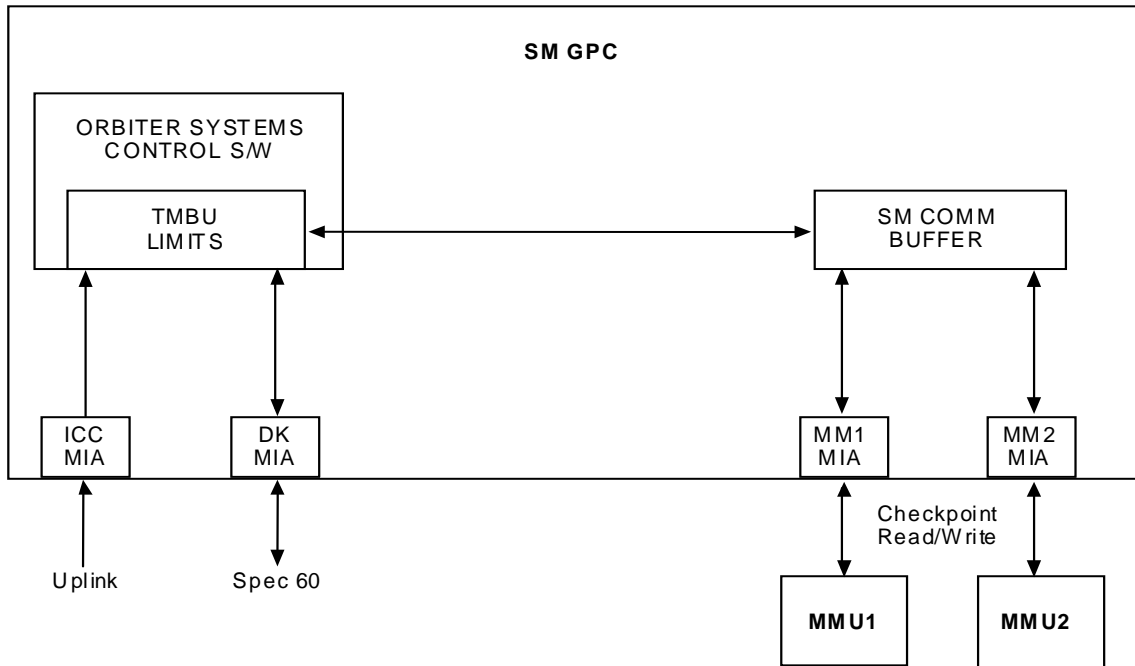
The term "*taking a checkpoint*" is used to describe the process of recording a block of information from the SM GPC's memory on the MMUs. This information includes things like the TMBU data, certain constants (used to regulate the timing of certain periodic orbiter systems events), and the TDRS state vector.

The TMBU data are such things as fault detection limits or annunciation inhibits for various parameters. These can be modified either by the crew (rarely) or the MCC via uplink (usually). Since these values tend to be changed from time to time during a normal mission, saving them on the MMUs makes it possible for the crew to recover all the changes easily, in the event of an SM GPC failure that requires loading the SM software into another GPC.

Checkpoints are written to both MMUs via item entries to SPEC 1 (DPS UTILITY) and SPEC 60 (TABLE MAINTENANCE). Upon completion of the checkpoint write to each MMU, the crew receives feedback as to its success or failure. In addition, since the checkpoint process uses the SM Common Buffer any attempt to write a checkpoint while the Common Buffer is in use for another process will result in a CM BUF BSY CRTX fault message (where X is the IDP/CRT being used to command a checkpoint write). Examples of other processes that use the SM Common Buffer are TFL loads and IDP loads via the SM GPC.

### **11.2 CHECKPOINT RETRIEVE**

Once written to the MMUs, the checkpoint can be retrieved by a new GPC being made the SM GPC (or the current SM GPC being PRO'ed to OPS 000 and back to SM OPS 201). A checkpoint retrieval likely would only occur in the event of the failure of the current SM GPC or a Deorbit Prep Backout. In order to retrieve the SM checkpoint from the MMUs, the new SM GPC must have prior authorization. This is given by arming ITEM 12 on SPEC 1 (DPS UTILITY). If this item is enabled when the new SM GPC transitions from OPS 000 to SM OPS 201, the checkpoint will be retrieved and the changes recorded in it will "overwrite" the corresponding nominal values in the SM software that came in with the SM OPS 2 overlay. Successful retrieval of the checkpoint automatically disables ITEM 12 on SPEC 1 (DPS UTILITY) until the crew re-enables it. Figure 11-1 shows the interfaces for an SM checkpoint.



td601a\_029.cnv

Figure 11-1. SM checkpoint data flow

## **12.0 MASS MEMORY INPUT/OUTPUT**

### **12.1 DESCRIPTION**

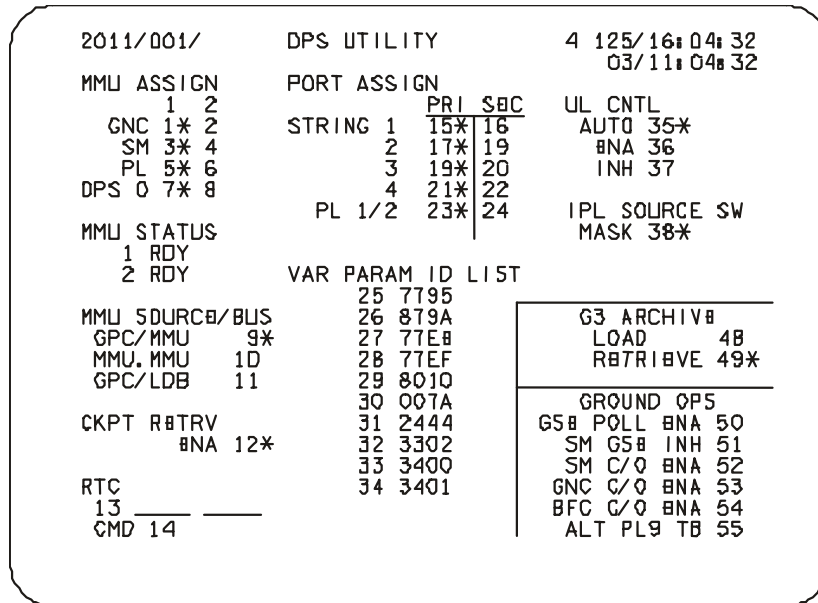
There are two Modular Memory Units (MMUs) on the orbiter. Each is a digital solid-state storage device. Each contains multiple copies of all the software for the GPCs. Each also contains orbiter systems data (SM checkpoint), which can be written to or read from the MMU as required.

Before an OPS transition is begun, the MMUs are both prepositioned to the beginning of the desired software files. If neither MMU completes the preposition, the OPS transition will be aborted, and the old OPS will continue to process.

### **12.2 MMU INSIGHT**

The items in the "MMU ASSIGN" area in the upper left-hand corner of SPEC 1 (DPS UTILITY) are used to control which of the MMUs will be used for various transactions. There are separate items for GNC, SM, PL, and OPS 0. In the case of an OPS transition or freeze dry, the active item for the major function involved in the OPS transition or the active item for OPS 0 (for the freeze dry case) specifies which MMU will be the source for the software loaded in for the new OPS. (Remember that both MMUs are prepositioned, and that if the selected MMU fails, the other one will be used to complete an OPS transition automatically. While there is no preposition for a freeze dry, if the selected one fails, the other will still be used to complete the overlay.) For accesses of the MMUs, other than OPS transitions and freeze drying, the active item for the major function involved specifies which MMU is used. There is no automatic switchover to the alternate MMU for any type of MMU access other than an OPS transition or a freeze dry.

Just below the MMU ASSIGN section on SPEC 1 (DPS UTILITY) is the "MMU STATUS" area (see Figure 12-1.) The indicators in this area give insight into the current status of each MMU. If a MMU is powered on and is not currently being accessed by a GPC, its indicator will read "RDY". If it is powered off or is currently being accessed by a GPC, its indicator will read "BSY". The GPCs are unable to distinguish between a MMU that is really busy from one that is powered off. That is why the fault message generated when attempting to access a MMU that is powered off or in use by another GPC reads "OFF/BSY MMUX".



TDE010121. CRT: 1

**Figure 12-1. SPEC 1, DPS UTILITY**

The next area encountered when continuing down the left-hand side of SPEC 1 (DPS UTILITY) is the "MMU SOURCE/BUS" area. The items in this area are used to control the source of software overlays during OPS transitions and which set of data buses are used. The normal selection is GPC/MMU. This item specifies that another GPC will be the source for the software overlayed during an OPS transition, if possible. If no other GPC already has the software, then the default source will be the MMUs. The data buses used for the transfer will be the MM data buses. If the "MMU/MMU" item is selected, it forces the software to come only from the MMUs, down the MM data buses. This item would only be used to force a "clean" load of GPC software from MMU, if the current load was thought to be contaminated in some way. If the "GPC/LDB" item is selected, it forces the software to come only from another GPC and down the LDB data buses. If no other GPC has the software requested, the OPS transition aborts and the old OPS continues to be processed. For this item to be used, it would be necessary for a number of orbiter failures to have occurred previously.

### 12.3 ROLL-IN SPECS

Since the memory size of a GPC is limited, the total amount of software that is desired to be loaded into the SM OPS 2 configuration for most flights is greater than the GPC's memory capacity. To solve this problem, certain SM OPS 2 displays are stored on the MMUs. These displays are called "roll-in SPECS", since a software capacity has been provided in SM OPS 2 to "roll-in" as many as two of them at one time from the MMU. There are two buffer spaces provided in the SM GPC software that are empty when SM is initialized and are used to hold roll-in SPECS as they are requested by the crew.

The process of calling such a SPEC is no different from the process of calling any other SPEC. However, a message is displayed to the crew while the SPEC's software is being retrieved from the MMU. Once two roll-ins have been called, both buffers are filled. Even if the roll-ins are RESUME'ed, they remain in the buffers and can be recalled without accessing the MMUs again. When a third roll-in is called, the software must select a buffer to write over in order to store it. The usual choice is the buffer currently being used by the roll-in called the longest ago, unless it is currently being displayed. However, if both of the current roll-ins are currently being displayed, then this prevents the software from using either buffer to store the request and the requested roll-in will not be retrieved from the MMU.

In general, the crew has no need to keep track of what roll-ins are currently in the buffers. The only time it might be a good idea to do so is when one MMU has failed. In that case, the crew would want to ensure that the two roll-ins that are most needed are kept in the buffers.

## 13.0 BFS ENGAGE

### 13.1 BFS ENGAGE OVERVIEW

As we have seen previously, the BFS has both GNC and SM software functions active simultaneously. The crew makes use of the BFS's SM software to monitor and control many orbiter onboard systems. Normally, the PASS software is used to monitor and control the GNC systems (those used to guide and control the vehicle in flight). The BFS follows the process of PASS control of GNC systems and monitors the data gathered by these systems. It does this so that it will always be ready to take over control of the total vehicle if the crew **engages the BFS**.

Why was the BFS developed in the first place? Why would the crew ever need to engage it? Originally, the fifth GPC was conceived as a computer dedicated to operating the orbiter systems (communications, electrical power, etc.) during the ascent and entry phases, with no capability to fly the vehicle. The PASS was designed from the beginning as a redundant system for control of flight. In the PASS, there are four GPCs running the same software and checking on each other continuously. Since each PASS GPC is controlling one-fourth of the flight control systems (i. e., one string), the failure of one PASS GPC should not affect vehicle control significantly. However, it was not the potential failure of a GPC, or any other piece of hardware, that made the development of a BFS engage capability necessary. Rather, it is the very nature of the PASS software. In a redundant set, such as is used during the ascent and entry phases of a shuttle flight, each of the PASS GPCs is **running exactly the same software**. Since the software is identical in each of the PASS GPCs, a coding error in the PASS software has the potential to cause the failure of all the PASS GPCs simultaneously. Thus, the BFS software is written by a different government contractor from the one that develops the PASS software. In addition, the algorithms implemented for each system are different. This makes the odds extremely low that the same coding error could exist simultaneously in both systems. So, the purpose for having a BFS is to take over control of the shuttle in the event of a PASS software failure.

It is also conceivable that engaging the BFS would become necessary in certain scenarios involving sequences of failures less catastrophic than a total failure of the PASS software. For example, communication between the GPCs and certain flight control and navigation sensors is essential for maintaining control of the vehicle. If communication was lost (i. e., "commfaulted") between PASS and all the units of one type of sensor (e. g., all three IMUs), then PASS could not continue to fly the vehicle safely. If the cause of at least one of the commfaults was the failure of a PASS GPC, then engaging the BFS would recover communication with the unit that the failed PASS GPC formerly commanded, allowing continuation of the mission.

It should be noted that if the BFS was engaged during the ascent portion of a mission, it is likely that the PASS GPCs can be re-IPL'ed and recovered for use on orbit and during entry. The software used in these phases of flight is substantially different from that used during ascent and likely does not contain the same coding error that made engaging the BFS necessary in the first place. The procedures used to **disengage** the BFS are both lengthy and complicated (and beyond the scope of these lessons) and can only be performed on orbit. Thus, if the BFS is engaged during the entry phase of flight, the BFS remains engaged until the mission is completed.



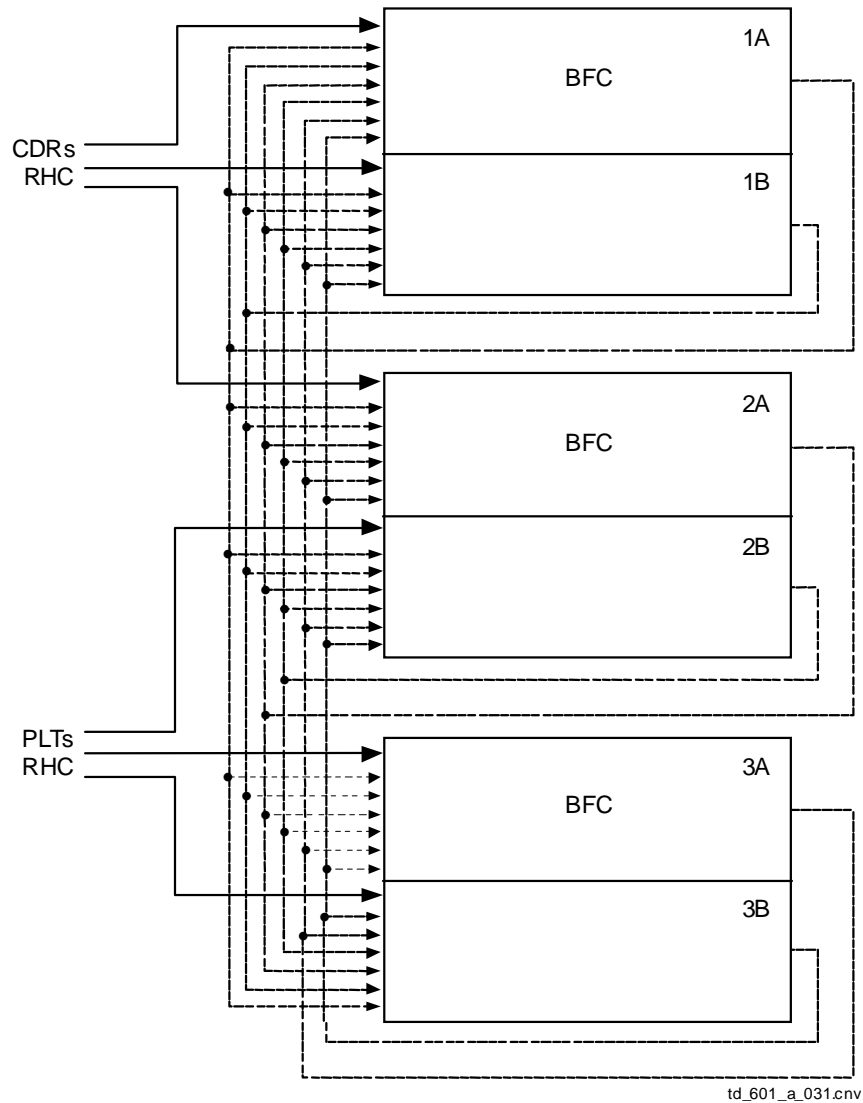
The decision to engage the BFS is not one that is made lightly. If the BFS has been engaged and the BFS GPC subsequently fails, there is no further GPC redundancy to maintain control of the vehicle. In addition, the software algorithms of the BFS are, of necessity, simpler than those of PASS, since the BFS is required to perform both SM and GNC functions simultaneously in the same memory space. The simpler algorithms of the BFS are not as effective as those in the PASS at managing and filtering the data from redundant sensor systems (to protect against hardware failures in those systems) or at handling off-nominal flight profiles and trajectories. Thus, the BFS is not as likely to be able to maintain vehicle control under certain conditions as would the PASS. (Two classic examples of this simplification are the lack of manual flight control capability during powered ascent and the lack of "autopilot" at any other time in the BFS.) In any event, the actual process of commanding a BFS engage is very simple. Either the Commander, the Pilot, or both has to press the BFS ENGAGE pb on top of their respective Rotational Hand Controller (RHC) (using eight pounds of pressure). The process is fully automatic after that.

## **13.2 BFS ENGAGE PROCESS**

### **13.2.1 Hardware**

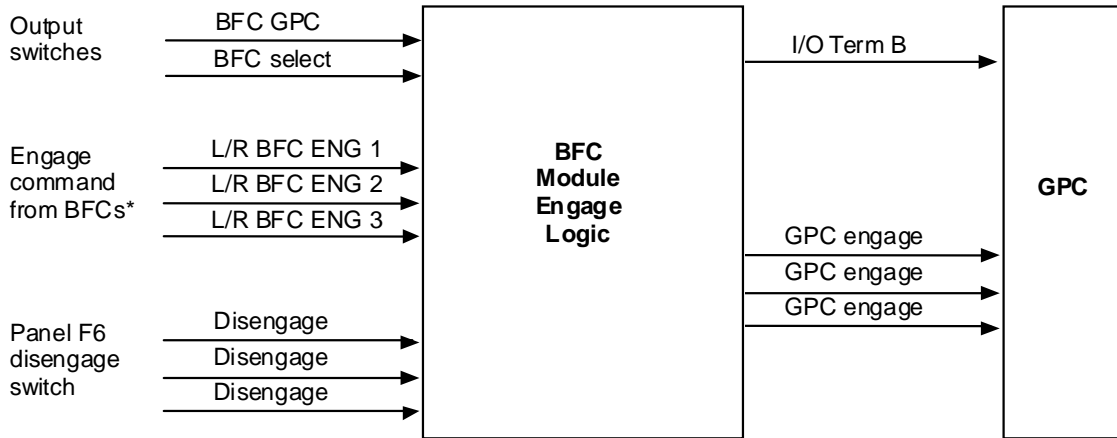
The mechanization of the BFS engage process requires the proper operation of a sequence of several hardware and software functions in order to complete successfully. Taking the hardware first, the process is initiated by the Commander or the Pilot by depressing the red pb on top of the RHC. Each pb has three contacts, which are powered from the left or right display driver unit (DDU) internal power supplies (each contact from a different power supply). These discrettes are routed (one each) to three of the BFC modules. The discrettes from the Commander's RHC are routed to BFC modules 1A, 1B, and 2A. The discrettes from the Pilot's RHC are routed to BFC modules 2B, 3A, and 3B. (Each BFC module, except module 3B, is associated with a specific GPC and will end up sending it several discrete input signals.)

The BFC modules are electrically separate units that are packaged two to a box with one box in each of avionics bays 1, 2, and 3. The main function of the BFC modules is to mediate the BFS engage process. The discrete that each BFC module receives from one of the RHCs is routed to the "front-end" of all the BFC modules (including itself), as can be seen in Figure 13-1. This allows an engage discrete from either RHC to be routed to all BFC modules, even though each RHC discrete is physically connected to only one BFC module. In other words, if either the Commander or the Pilot initiates an engage, all the BFC modules know about it.



**Figure 13-1. BFS engage pushbutton interface to the BFS modules**

Once all of the BFC modules have received the signals from those BFC modules that are connected to the RHC pushbutton, each module processes the discretes individually. How each one uses the signals to generate discretes to be sent on to the GPCs depends upon the setting of the OUTPUT switch for the GPC associated with each module. These switches are wired in such a manner as to send a "0" or "1" value for two discretes (BFC GPC SELECT and BFC SELECT OFF) to each BFC module's engage logic. Taken together, the states of these two discretes determine the states of the discretes (the three GPC ENGAGE discretes and I/O TERM B) that each BFC module will, in turn, send to its associated GPC when the crew presses either of the engage pbs. Figure 13-2 shows an overview of the input and output discretes to/from the engage logic of a typical BFC module.

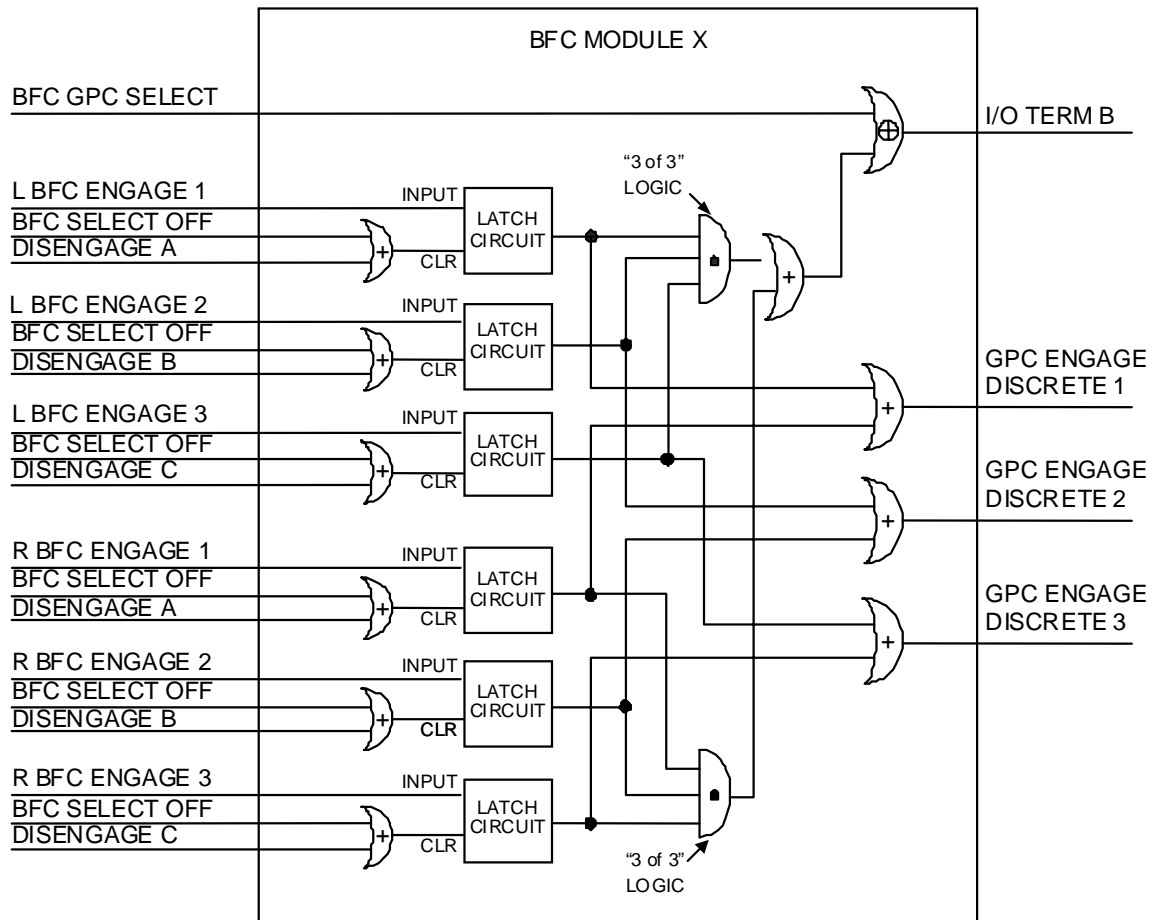


\*Originally from RHCs

td601a\_032.cnv

**Figure 13-2. BFC input/output overview**

The BFC SELECT OFF discrete is set to "1" for all the BFC modules whenever *all* GPCs are in NORMAL or TERMINATE. This is the normal configuration on orbit and is designed to prevent an accidental engage, when there is no GPC with its OUTPUT switch in the BACKUP position. The BFC SELECT OFF discrete is also set to "1" for any GPC with its OUTPUT switch in TERMINATE. The reason for this is that the BFC module associated with any GPC in TERMINATE will also send an I/O TERM B discrete to that GPC, preventing it from transmitting on the flight criticals. Any GPC that cannot transmit on the flight criticals certainly could not interfere with the BFS taking over control of the vehicle during an engage. Thus, the general purpose of the BFC SELECT OFF discrete is to prevent a given BFC module from causing its associated GPC to respond to an engage.



td601\_033.cnv

**Figure 13-3. BFC engage logic**

The BFC GPC SELECT discrete is set to "1" for the BFC module associated with the GPC with its OUTPUT switch in BACKUP. (The wiring of the OUTPUT switches takes care of the case where more than one GPC's OUTPUT switch is in BACKUP. Only the BFC module for the highest numbered GPC with its OUTPUT switch in BACKUP has its BFC GPC SELECT set to "1".) This is the normal configuration for ascent or entry and sets up the module to be ready to respond to an engage request from the crew. The BFC GPC SELECT discrete is also set to "1" for any GPC with its OUTPUT switch in TERMINATE. The reason that this is done has to do with the logic used to generate the I/O TERM B discrete in the BFC modules. A GPC in TERMINATE will continue to receive its I/O TERM B discrete set to "1" after an engage.

The BFC modules also can receive three additional discretes from the BFS DISENGAGE switch on Panel F6. These are the Disengage A, B, and C discretes. Their purpose is to cause the BFC modules to change the discretes that each sends to its associated GPC, forcing the BFS to return to a pre-engaged state and forcing PASS to take back control of the vehicle. This will be covered in greater detail later in this section.

As was mentioned earlier, the state that each BFC module sets the discrettes that it sends to its associated GPC depends upon the state of the discrettes that are sent to the BFCs. For this discussion, we will refer back to Figure 13-3. In this functionally equivalent drawing (i. e., not the real circuit, but a simpler one that works the same way), "logic gates" have been used to represent the function of the BFCs' electronics.

There are three types of logic gates represented. The first type is an "OR" gate, which is pictured symbolically somewhat like a "crescent moon" with a plus sign in it. In an OR gate, if any one of the inputs (discrettes entering from the left) or both of the inputs are set to "1", then the output (the discrete exiting on the right) is also set to "1". The second type is an "AND" gate, which is pictured symbolically like a "half-moon" with a dot in it. In an AND gate, if **all** of the inputs are set to "1", then and only then is the output set to "1"; otherwise, it is set to "0". The third type is an "EXCLUSIVE OR" gate, which is pictured symbolically like a "crescent moon" with a circled plus sign in it. This type of gate differs from the OR gate only in the special case where both of its inputs are set to "1". The EXCLUSIVE OR gate will set its output to a "0" in this case. (In other words, "either A or B, but not both".) A summary of the function of the logic gates can be seen in the Truth Tables shown below.

Exclusive OR			OR			AND			
Inputs		Output	Inputs		Output	Inputs			Outputs
0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	1	1	0	0	0
0	1	1	0	1	1	0	1	0	0
1	1	0	1	1	1	0	0	1	0
						1	1	0	0
						0	1	1	0
						1	0	1	0
						1	1	1	1

The other circuit of import to us in this discussion is the LATCH, represented symbolically here as a box. This circuit is used to "trap" the condition when its input state is set to "1" and hold its output equal to "1", until such time as the "CLR" input is set to "1", when it will "clear" its output to "0". The need for this circuit is self-evident. If it was not there, then the crew would have to continually hold down the engage pbs on their RHCs in order to maintain the engaged state. That would be very difficult. Instead, a single "press and release" of either engage pb will set the associated latches in each module, electronically "holding down the pb" for the crew.

Let's assume a nominal DPS config for ascent or entry, with GPCs 1 - 4 processing PASS and with their OUTPUT switches in NORMAL. GPC5 is processing BFS, and its OUTPUT switch is in BACKUP. Thus, the BFC modules for all of the PASS GPCs have both their BFC GPC SELECT and their BFC SELECT OFF discrettes set to "0" (because of the way the OUTPUT switches are wired). Prior to the BFS being engaged, each of these modules would be sending an I/O TERM B discrete set to "0" and all three engage discrettes also set to "0" to its associated PASS GPC. The BFC module for the BFS GPC would have its BFC GPC SELECT discrete set to "1" and its BFC SELECT OFF discrete set to "0". Thus, it would set the I/O TERM B discrete to "1" and the engage discrettes to "0" for the BFS GPC. The important point to remember is that the logic circuits are always active and always generating either a "1" or a "0" output for all of the discrettes sent to the GPCs. Thus, it is only the differing inputs from the GPC OUTPUT switches that determine what state the discrettes sent from the BFC modules to their associated GPCs will be. Likewise, it is the change in state of those all-important inputs to the BFCs, the L&R BFC ENGAGE 1/2/3 discrettes that changes the state of the BFC modules' outputs at the time of engage.

As we have seen in Figure 13-1, when either the Commander or the Pilot (or both) press the engage pb on their respective RHCs, the discrettes are routed through three of the modules and "strapped back" to all six modules. As can be seen by examining the logic circuits in Figure 13-3, pressing either engage pb will set the inputs to three of the LATCH circuits. These circuits will trap the "1" and output it until either the BFC SELECT OFF or the associated DISENGAGE discrete is set to "1", which will clear the LATCH. Remember that the BFC SELECT OFF discrete is only set to a "1" when **all** GPCs have their OUTPUT switches in NORMAL or for a GPC with its OUTPUT switch in TERMINATE, due to the wiring of the GPC OUTPUT switches. Either of these situations is normally only encountered on orbit, with the express intention of preventing any engage. So, the BFC SELECT OFF is not likely to be a "1". The DISENGAGE discrettes would only be present when the BFC DISENGAGE switch on Panel F6 is toggled; so these discrettes would likely be "0" as well.

Since there would be no inputs to clear the LATCHes, they would output "1"s to the GPC ENGAGE discrete drivers and the "3 of 3" logic. This "3 of 3" logic is represented in Figure 13-3 by the 3-input AND gate. If all three inputs are not a "1", then the output remains "0". This prevents the I/O TERM B discrete from being set to "1". As we will see later, this prevents the PASS GPC in question from "recognizing" the engage. The purpose of this circuit is to act as a protective filter to prevent a GPC from "accidentally" recognizing an incorrect set of discrettes (due to a failure of one or more sets of electronics along the circuit path) as a BFS engage. As can also be seen in Figure 13-3, if either of the engage pbs is pushed, then one or the other of the "3 of 3" logic circuits will output a "1" to the OR gate downstream of them. Since an OR gate outputs a "1" when **either** input is a "1", then a BFS engage from either RHC pb is the same. However, the "3 of 3" logic circuits for each set of L or R BFC ENGAGE discrettes are separate, e. g., it is not possible for two discrettes from the left and one from the right to equal an engage.

The above discussion with respect to changes in the discrete states of the BFC modules applies to that module associated with the BFS GPC, as well. The point in the circuit path where the difference emerges is the logic gate in the upper right corner of Figure 13-3, the EXCLUSIVE OR gate. Remember that this type of gate is identical to a regular OR gate, except for the case where both inputs are set to "1". In this unique case, an EXCLUSIVE OR gate will output a "0". For the BFC modules associated with the PASS GPCs, the BFC GPC SELECT discrete is set to "0"; for the module associated with the BFS GPC, it is set to "1". Thus, when the "3 of 3" logic circuits and the OR gate downstream of them drive a "1" as one of the inputs to the EXCLUSIVE OR gate, it is the state of the other input (the BFC GPC SELECT discrete) that controls the output. For the BFC modules associated with the PASS GPCs, the BFC GPC SELECT discrete set to "0" makes the output of the EXCLUSIVE OR gate be a "1". For the BFC module associated with the BFS GPC, the BFC GPC SELECT discrete is set to "1". Since this makes both inputs to the EXCLUSIVE OR gate a "1", it outputs a "0". Thus, PASS GPCs get an I/O TERM B discrete high and the BFS GPC gets an I/O TERM B low. Since the I/O TERM B discrete set to "1" inhibits a GPCs transmitters on the flight critical data buses, such a GPC can no longer control the vehicle. Therefore, after an engage, PASS GPCs no longer have the hardware capability to control the vehicle, and the BFS GPC does. In addition, as we will see in the next section, there are major changes in the functions of the software accompanying these hardware discrete state changes.

### 13.2.2 Software

The system software for PASS GPCs has a special function to deal with a BFS engage. Any PASS GPC that is in command of a flight critical data bus will suddenly become unable to transmit over the flight criticals when its I/O TERM B discrete goes to a "1". Since the transmission does not get out of the inhibited GPC transmitter, the MDM it was intended for never receives a transmission. Since the MDM was not polled, it does not respond. The PASS GPCs notice this and log an "I/O error". Whenever a PASS GPC logs an I/O error against a flight critical data bus transaction, the software branches to a special routine that checks to see if the I/O TERM B and **all three** of the GPC ENGAGE discretetes are set to "1". If this is true, the software branches to a routine that lights the CAM diagonal light for this GPC and then shuts down software processing; i. e., "fails to software halt". Since all four of the PASS GPCs are running in a redundant set, they all do this simultaneously. The characteristic signature is all four CAM diagonal lights lit. (In addition, there is cyclic software in the PASS to cause GPCs that were not commanding a flight critical and not in the redundant set (e. g., SM GPC on orbit) to "recognize" the engage. For such a GPC, this mainly amounts to relinquishing control of the second CRT specified by the BFC CRT SEL switch. Such a GPC does not go to software halt.)

The system software of the BFS GPC is cyclically checking the status of the I/O TERM B and GPC ENGAGE discretes. If the software detects the I/O TERM B and **any two** of the GPC ENGAGE discretes set to "1", the BFS changes state from pre-engaged to engaged. The reason that the BFS software's engage criteria is less stringent than that of PASS is that, upon engage, it is **imperative** that the BFS take over control of the vehicle, since the PASS is going to software halt. Thus, it would be undesirable to allow the failure of a single GPC ENGAGE discrete driver, for instance, to prevent the BFS succeeding in engaging. When the BFS changes its software state to engaged, a quantity of software that was not previously executing begins to do so. For example, the BFS begins to transmit commands to the equipment on the flight critical data buses, just like PASS was doing before the engage. It automatically starts to drive the second CRT specified by the position of the BFC CRT SEL switch. (If the BFC CRT DISPLAY switch was in OFF at the time of the engage, the BFS will start driving CRTs 1 and 2.) If use of the third and fourth DPS displays is desired by the crew, they can be assigned. The BFS has the capability to drive all four displays simultaneously. An additional indication of successful BFS engage is the steady (i. e., not flashing) illumination of the BFC lights on the two "eyebrow" panels.

### 13.2.3 Failures

As can be inferred from the design of the circuitry we have discussed in this section, a number of things could go wrong during a BFS engage. For example, any one of the gates in the BFC module for a given GPC that drive the I/O TERM B and GPC ENGAGE discretes (or the associated receivers in the GPC) could have a failure. This could be as insignificant as the failure of one of the GPC ENGAGE discretes to a "0" for the BFS (resulting in a successful engage, due to the BFS software's less stringent requirements), to the same failure at a PASS GPC (resulting in that one GPC not going to software halt and dual commanders on the string commanded by that GPC), to the I/O TERM B discrete being failed to a "1" at the BFS GPC (resulting in loss of control of the vehicle). (While a failure of the I/O TERM B discrete to a "1" has never been seen in-flight, a failure of this discrete to "0" for the BFS GPC was observed during two flights. The location of the failure was found to be a broken connector and contamination in one of the wiring harnesses that carry the discrete signals between the switches, BFC, and GPC.) Even though failure modes can be conjectured, the circuitry of the BFC modules is considered very reliable. In fact, only two real-world failures of a BFC module circuit element have ever been confirmed either in-flight or in ground tests in the history of the Shuttle program, and these were early in the program.



### 13.3 BFS DISENGAGE PROCESS

If the problem that resulted in engaging the BFS occurred during ascent, then it should be possible to recover operation of the PASS software once a safe orbit is achieved (using the engaged BFS for vehicle control). This is desirable, since the BFS has no payloads software and no capability to align the orbiter's IMUs, meaning a normal mission could not be accomplished using it alone. The crew procedures for this process are extensive and complicated. We will not attempt to explain them here. In essence, the software in the PASS GPCs is dumped, using the hardware dump capability of the GPCs. (Since this function requires putting the GPC being dumped in TERMINATE, this sets the BFC SELECT OFF discrete to a "1". This, in turn, causes the BFC module to "forget" about the engage. The procedures have the crew press the ENGAGE pb over again to reset the discrettes from the BFC to the GPC.) The dumps are done so that MCC can analyze the failure mechanism that resulted in the BFS engage and design a software fix to prevent its recurrence during entry. After that, the PASS GPCs are IPL'ed to reload them with usable software. One PASS GPC is loaded with SM OPS 2 in order to monitor the orbiter systems more effectively. The other three PASS GPCs are loaded with GNC OPS 3 software in order to be ready to do an entry using PASS.

When the time comes to give control back to the PASS, the crew toggles the BFC DISENGAGE switch to momentarily set the DISENGAGE A/B/C discrettes to "1". As can be seen in Figure 13-3, this clears the LATCH circuits and returns the outputs of each BFC module to the state each was in prior to the BFS engage. When the PASS GPCs detect the I/O TERM B and the GPC ENGAGE discrettes returning to "0", the PASS resumes control over the flight criticals. (The engage discrettes are redundancy-managed across the common set, in order to eliminate "confusion", in the event of a discrete being failed to a "1".) When the BFS receives the I/O TERM B set to "1", its hardware inhibits transmission over the flight criticals. This results in eight fault messages of the I/O ERROR FFX and I/O ERROR FAX varieties. When this happens, the BFS detects the changed state of its discrettes and resumes pre-engaged tracking of PASS. However, the BFS software does not release all of the software modules it started at engage, and may issue some spurious fault messages. Prior to depending on the BFS to back up the PASS for entry, it will be PRO'ed to OPS 0 and cycled to HALT and back to RUN, in order to restart only normal BFS pre-engaged software processing.

**APPENDIX A**  
**ACRONYMS AND ABBREVIATIONS**

ACK	Acknowledge
ADC	Analog-to-Digital Converter
BBU	Battery Backup Unit
BCE	Bus Control Element
BFC	Backup Flight Controller
BFS	Backup Flight System
bp	barberpole
BSL	Backup System Loader
BSY	Busy
BTU	Bus Terminal Unit
C&W, C/W	Caution and Warning
CAM	Computer Annunciation Matrix
CPU	Central Processing Unit
CRT	Cathode Ray Tube
DDU	Display Driver Unit
DISP	Display function
DK	Display/Keyboard (bus)
DPS	Data Processing System
ECC	Error Correcting Code
EEPROM	Electrically Erasable Programmable Read Only Memory
EIU	Engine Interface Unit
ENA	Enable
EXEC	Execute
FA	Flight critical Aft (MDM)
FAULT SUMM	Fault Summary (display)
FC	Flight Critical (bus, MDM)
FDF	Flight Data File
FF	Flight critical Forward (MDM)
GMT	Greenwich Mean Time
GNC	Guidance, Navigation, and Control
GPC	General Purpose Computer
GSE	Ground Support Equipment
H/W	Hardware
HDR	High Data Rate
HEX	Hexadecimal (base 16 numbers)

I-LOADs	Initial Loads (pre-launch default values)
I/O	Input/Output
IC, ICC	Inter-Computer Communication (buses)
IDP	Integrated Display Processor
IFM	In-Flight Maintenance
IMU	Inertial Measurement Unit
IOP	Input Output Processor
IP	Instrumentation/Pulse Code Modulation Master Unit (bus)
IPL	Initial Program Load
KYBD, KBU	Keyboard Unit
LDB	Launch Data Bus
LDR	Low Data Rate
LPS	Launch Processing System
MAJ FUNC	Major Function switch
MC	Memory Configuration
MCC	Mission Control Center
MDM	Multiplexer/Demultiplexer
MDU	Multifunction Display Unit
MEDS	Multifunction Electronic Display System
MET	Mission Elapsed Time
MF	Major Function
MIA	Multiplexer Interface Adapter
MLS	Microwave Landing System
MM	Major Mode, Mass Memory (bus)
MMU	Modular Memory Unit
MSG	Message
MTU	Master Timing Unit
NAV	Navigation
NBAT	Nominal Bus Assignment Table
NSP	Network Signal Processor
OI	Operational Instrumentation (MDM, bus)
OPS	Operational Sequence, Operations
OTP	Operational Test Program
PASS	Primary Avionics Software System
pb	push button
pbi	push button indicator
PCMMU	Pulse Code Modulation Master Unit
PDI	Payload Data Interleaver
PF	Payload Forward (MDM)
PL	Payload

PNL	Panel
PRI	Primary
PRO	Proceed
PROM	Programmable Read-Only Memory
PWR	Power
RAM	Random Access Memory
RCVR	Receiver
RDY	Ready
RHC	Rotational Hand Controller
RM	Redundancy Management
RMS	Remote Manipulator System
RTC	Real Time Command
RTLS	Return To Launch Site (abort)
S/W	Software
SEC	Second, Secondary
SM	Systems Management
SPEC	Specialist function
SPL	Scratch Pad Line
SRB	Solid Rocket Booster
SSL	System Software Loader
SSME	Space Shuttle Main Engine
SST	Single System Trainer
STBY	Standby
STP	Self Test Program
STRG	String
sw	switch
sync	synchronization, synchronize
SYS	System
SYS SUMM	System Summary (display)
tb	talkback
TDRS	Tracking and Data Relay Satellite
TFL	Telemetry Format Load
TMBU	Table Maintenance Block Update
UL	Uplink
WIX	Wait for Index
XMTR	Transmitter

# TRAINING MATERIALS EVALUATION

Please answer the following questions regarding the lesson you just completed. Your feedback will allow *us to produce more effective training materials*. When complete, mail to: **Space Shuttle DPS/Navigation Department, DT35.**

**TITLE/CODE OF LESSON:** DPS Familiarization Workbook/DPS FAM 21002

**SIZE OF AUDIENCE/CLASS:**

## 1. How well did this lesson meet its purpose?

For each statement below, mark one box on the scale:

- a. The lesson objectives are clearly stated.
- b. The lesson objectives are clearly defined.
- c. The lesson effectively teaches skills and information.
- d. The lesson meets its purpose and objectives.

Strongly Disagree			Strongly Agree		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## 2. How satisfying is the content of this lesson?

For each statement below, mark one box on the scale:

- a. The information is structured in a logical flow.
- b. The content is clear.
- c. The content is complete.
- d. The level of detail is correct for this information.
- e. The amount of information is effective.
- f. The graphics contribute to my understanding.

Strongly Disagree			Strongly Agree		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## 3. How appealing was the presentation of this lesson?

For each statement below, mark one box on the scale:

- a. The overall presentation is appealing.
- b. The visuals chosen are appropriate for the lesson.
- c. The visuals make the information more interesting.
- d. The graphics are legibly reproduced.
- e. The audio/visual or print quality is good.

Strongly Disagree			Strongly Agree		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## 4. How valuable is this information?

For each statement below, mark one box on the scale:

- a. The lesson teaches skills and information I need.
- b. The lesson meets my expectations.
- c. This information is useful for later reference.
- d. I would recommend this lesson to others.

Strongly Disagree			Strongly Agree		
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**PLEASE WRITE YOUR COMMENTS/QUESTIONS ON THE BACK OF THIS FORM.  
EXPLAIN ANY NEGATIVE ANSWERS IN SPECIFIC TERMS  
THANK YOU IN ADVANCE FOR YOUR ASSISTANCE!**