



National Aeronautics and  
Space Administration

NASA CR-188254

Lyndon B. Johnson Space Center  
Houston, Texas 77058

**IMPLEMENTING THE SPACE SHUTTLE DATA  
PROCESSING SYSTEM WITH THE SPACE  
GENERIC OPEN AVIONICS ARCHITECTURE**

N94-23571

Unclass

G3/16 0202109

December 1993

John R. Stovall  
Richard B. Wray

Prepared by:

Lockheed Engineering & Sciences Company  
Houston, Texas

Job Order 60-911  
Contract NAS 9-17900

for

FLIGHT DATA SYSTEMS DIVISION  
JOHNSON SPACE CENTER

(NASA-CR-188254) IMPLEMENTING THE  
SPACE SHUTTLE DATA PROCESSING  
SYSTEM WITH THE SPACE GENERIC OPEN  
AVIONICS ARCHITECTURE (Lockheed  
Engineering and Sciences Co.) 43 p

LESC-30753

)

.

)

.

)

**IMPLEMENTING THE SPACE SHUTTLE DATA  
PROCESSING SYSTEM WITH THE SPACE  
GENERIC OPEN AVIONICS ARCHITECTURE**

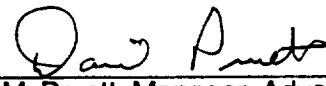
December 1993

Richard B. Wray, Advanced Systems Engineering Specialist  
John R. Stovall, Advanced Systems Engineering Specialist

APPROVED BY:



G.L. Clouette, Project Integration Specialist  
Flight Data Systems Department



D.M. Pruett, Manager, Advanced Programs  
Flight Data Systems Division

Prepared by:

Lockheed Engineering & Sciences Company  
Houston, Texas

Job Order 60-911  
Contract NAS 9-17900

for

FLIGHT DATA SYSTEMS DIVISION  
JOHNSON SPACE CENTER

LESC-30753

)

.

.

)

.

)

## PREFACE

This document has been produced by Mr. John R. Stovall and Mr. Richard B. Wray of Lockheed Engineering and Sciences Company (LESC), the codevelopers of the avionics architectures and standards being tailored in this document. The contributions of Mr. Ben Doeckel of LESG who participated in early development of the concepts for the avionics architectures, standard and tailorings represented in this document is acknowledged. Special acknowledgment is also given to Mr. Dave Pruett of the Johnson Space Center (JSC) for his support of the Advanced Architecture Analysis, assistance in the development of the avionics architecture and constructive criticisms of the proposed standard.

PRECEDING PAGE BLANK NOT FILMED

FILED IN 100-100000-1000

## DOCUMENT CHANGE RECORD

The following table summarizes the change activity associated with this document.

<b>ISSUE AND DATE</b>	<b>PAGE NO.</b>	<b>CHANGE SUMMARY</b>	<b>SECTION</b>

# CONTENTS

Section	Page
1. INTRODUCTION .....	1-1
2. SPACE SHUTTLE DPS SUMMARY .....	2-1-1
2.1 <u>REQUIREMENTS</u> .....	2-1-1
2.1.1 MISSION DERIVED .....	2-1-1
2.1.2 VEHICLE DERIVED.....	2-1-3
2.2 <u>DATA PROCESSING SYSTEM DESCRIPTION</u> .....	2-2-1
2.2.1 DATA BUSES .....	2-2-1
2.2.2 GPC OPERATIONS .....	2-2-1
2.2.3 MASS MEMORY UNIT .....	2-2-4
2.2.4 MULTIPLEXER/DEMULTIPLEXER UNIT.....	2-2-6
2.2.5 DISPLAY ELECTRONIC UNIT.....	2-2-6
2.2.6 TIME MANAGEMENT.....	2-2-6
2.3 <u>SHUTTLE SOFTWARE ARCHITECTURE</u> .....	2-3-1
2.3.1 APPLICATION PROCESSING SOFTWARE.....	2-3-1
2.3.2 SYSTEM SERVICES SOFTWARE.....	2-3-1
3. APPLICATION OF THE SGOAA TO THE SPACE SHUTTLE DPS.....	3-1-1
3.1 <u>HARDWARE ARCHITECTURE</u> .....	3-1-1
3.2 <u>INTERNAL HARDWARE ARCHITECTURE</u> .....	3-2-1
3.3 <u>SOFTWARE ARCHITECTURE</u> .....	3-3-1
3.3.1 SGOAA DATA SYSTEM SERVICES REQUIREMENTS.....	3-3-1
3.3.2 SPACE SHUTTLE ONBOARD FUNCTIONAL SOFTWARE ARCHITECTURE COMPARED TO THE SGOAA.....	3-3-2
4. CONCLUSIONS.....	4-1

## TABLES

<b>Table</b>	<b>Page</b>
3.2-1 Architectural Interface Classes .....	3.2-3

## FIGURES

<b>Figure</b>	<b>Page</b>
2.1-1 Space Shuttle Data Processing System Architecture.....	2.1-2
2.2-1 Space Shuttle General Purpose Computer (AP-101S) Data Bus Architecture.....	2.2-2
2.2-2 Space Shuttle General Purpose Computer Internal Block Diagram.....	2.2-3
2.2-3 Space Shuttle Mass Memory Unit Functional Block Diagram.....	2.2-5
2.2-4 Space Shuttle Multiplexer/Demultiplexer Block Diagram.....	2.2-7
2.3-1 Space Shuttle Primary Avionics Software Functional Structure.....	2.3-2
3.1-1 The Generic System Architecture Model Enables Flexible Use of Standards .....	3.1-2
3.1-2 Generic Processing External Hardware Architecture Model and Interfaces Are Adaptable .....	3.1-3
3.2-1 The Generic Processing Internal Hardware Architecture Can Be Tailored.....	3.2-2
3.2-2 The Generic Avionics Architecture Can be Applied to the Space Shuttle Processing Elements .....	3.2-4
3.2-3 The Generic Avionics Architecture Can be Applied to the Space Shuttle Processing Support Elements.....	3.2-5
3.2-4 The Generic Avionics Architecture as Applied to Space Shuttle Other Processing Elements.....	3.2-7
3.3-1 Space Shuttle Primary Avionics Software If Structured Based on the SGOAA Model .....	3.3-3
3.3-2 Space Shuttle Data Processing System Functions Can Be Tailored from the SDSS.....	3.3-4



## ACRONYMS

AGE	Aerospace Ground Equipment
BFS	Backup Flight System
BITE	Built-In Test Equipment
C&W	Caution & Warning
CCU	Computer Communications Umbilical
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
CRT	Cathode-Ray Tube
DDU	Display Driver Unit
DEU	Display Electronics Unit
DK	Display Keyboard
DPS	Data Processing System
DU	Display Unit
E	Effector
EEPROM	Electrically Erasable Programmable Read Only Memory
EIU	Engine Interface Unit
EP	Embedded Processor
EP(e)	EP Effector
EP(s)	EP Sensor
FC	Flight Critical
FCOS	Flight Computer Operating System
GAP	General Avionics Processor
GAP(M)	GAP used primarily as a multiplexer/demultiplexer
GAP(S)	GAP used for standard general purpose use
GMT	Greenwich Mean Time
GN&C	Guidance, Navigation and Control
GPC	General Purpose Computer
HAL-S	High-Order Software Language for Shuttle
ICC	Intercomputer Communication Data Bus
I/F	Interface
I/O	Input/Output
IOP	Input/Output Processor
IP	Instrumentation/PCM Master Unit Data Bus
JSC	Johnson Space Center
LB	Launch/boost
LESC	Lockheed Engineering & Sciences Company
LPS	Launch Processing System

MC	Memory Configuration
MCIU	Manipulator Controller Interface Unit
MDM	Multiplexer/demultiplexer
MEC	Master Events Controller
MET	Mission Elapsed Time
MIA	Multiplexer Interface Adapter
MM	Mass Memory Data Bus
MMU	Mass Memory Unit
MTU	Master Time Unit
NASA	National Aeronautics and Space Administration
OPS	Operational Sequence
OS	Operating System
PCM	Pulse Code Modulation
PCMMU	PCM Master Unit
PL	Payload
PASS	Primary Avionics Software System
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RMS	Remote Manipulator System
RTE	Run Time Environment
S	Sensor
SAE	Society of Automotive Engineers
SAP	Special Avionics Processor
SATWG	Strategic Avionics Technology Working Group
SCU	Sequence Control Unit
SGOAA	Space Generic Open Avionics Architecture
T&C	Test and Checkout

## **REFERENCES**

- [BUS85] Buskirk, Glenn, "AP-101S Space Shuttle GPC Design Handbook", IBM 85-C67-005, November 1985.
- [DPS2102] "Data Processing System (Hardware and Systems Software) Training Manual, DPS HW/SW 2102, NASA, April 1986
- [HAN89] Hanaway, J. F. and Moorehead, R. W., "Space Shuttle Avionics System", NASA SP-504, 1989.
- [STO93] Wray, R. B. and Stovall, J. R., "Space Generic Open Avionics Architecture (SGOAA) Standard", Job Order 60-911, Contract NAS9-17900 for the JSC, NASA CR-188269, LESC-30354-B, December 1993.
- [MACUNK] Macina, A. J., "Space Shuttle System Overview (Part 1)" Presentation, IBM
- [WRA93] Wray, R. B. and Stovall, J. R., "Space Generic Open Avionics Architecture (SGOAA) Reference Model Technical Guide", Job Order 60-911, Contract NAS9-17900 for the JSC, NASA CR-188246, LESC-30347-A, April 1993.

)

)

)

## 1. INTRODUCTION

This paper presents an overview of the application of the Space Generic Open Avionics Architecture (SGOAA) to the Space Shuttle Data Processing System (DPS) architecture design. This application has been performed to validate the SGOAA, and its potential use in flight critical systems. The SGOAA has been proposed as an avionics architecture standard with the National Aeronautics and Space Administration (NASA), through its Strategic Avionics Technology Working Group (SATWG) and is being considered by the Society of Automotive Engineers (SAE) as an SAE Avionics Standard. This architecture was developed for the Flight Data Systems Division of the NASA JSC by the LESC, Houston, Texas. This architecture includes a generic system architecture for the entities in spacecraft avionics, a generic processing external and internal hardware architecture, a six class model of interfaces and functional subsystem architectures for data services and operations control capabilities.

The SGOAA is documented in Reference [STO93] for the proposed standard and in Reference [WRA93] for the technical guide for the proposed standard. References [HAN89], [DPS2102], [MACUNK] and [BUS85] present a discussion of the Space Shuttle Avionics System. Section 2 provides an overview of the requirements, design and operation of the Space Shuttle avionics. Section 3 provides an overview of the tailoring of the SGOAA to the needs of the shuttle avionics. Section 4 provides some conclusions.

—

—

—

## **2. SPACE SHUTTLE DPS SUMMARY**

The architecture of the Space Shuttle avionics is shown in Figure 2.1-1. It consists of sensor and effector devices, general purpose hardware and software, and specialized hardware and software. The primary avionics software system (PASS) is the principal software used to operate the shuttle. This software contains all the applications and services code needed to fly the vehicle through all launch and orbit phases, and to manage the vehicle and payload while in orbit.

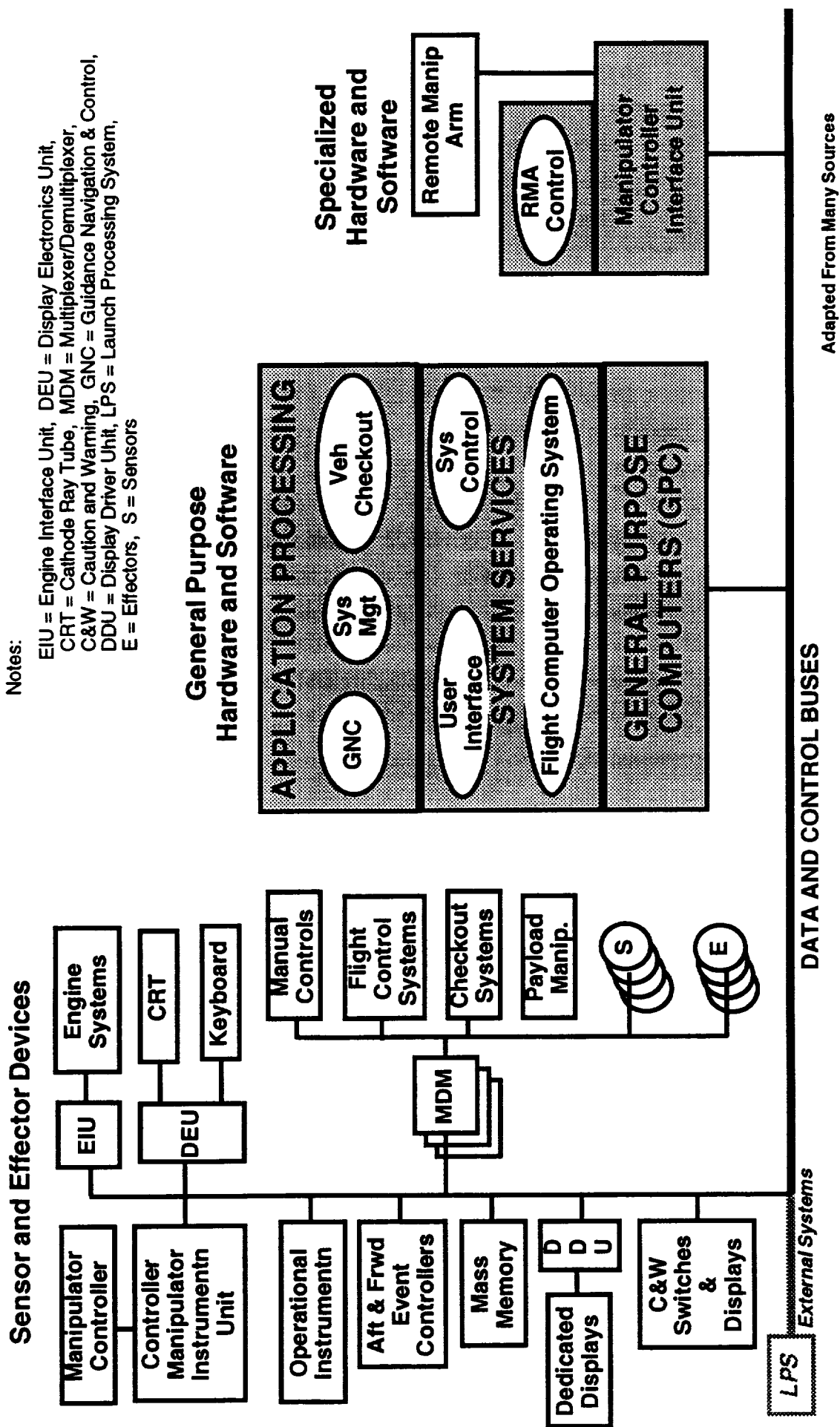
The key requirements on the DPS and the PASS are identified below. To meet these requirements, the four common (of the five) general purpose computers (GPC) are loaded with the same PASS code to perform the guidance, navigation and control functions simultaneously, with results compared. The fifth GPC contains a different set of software developed by a different company to take over vehicle control if the PASS code should have a generic design error. The software in this fifth computer is the backup flight system (BFS). It is only needed in critical flight phases such as ascent and descent. During less dynamic phases, different parts of the PASS software can be loaded into the four GPCs (allocated to the PASS) to support orbit activities. The GPC architecture and the software architecture, are summarized below to clarify the goals of the SGOAA tailoring described in Section 3.

### **2.1 REQUIREMENTS**

High level requirements for the Space Shuttle are summarized below. The requirements fall into two areas: those which are derived from the needs to perform the mission and those which are derived from the needs to safely control the vehicle.

#### **2.1.1 MISSION DERIVED**

- At least two safe methods of return to earth must be provided.
- An abort after one failure is not acceptable, therefore: fail op/fail safe is imposed. This dictates three strings to detect a failure and a backup string to recover; thus at least four strings are required
- Autonomous operation (onboard access for analysis of data) is required.
- Use of operational data to detect and isolate failures is required.
- Automatic failure detection and recovery for time critical functions is required.



Adapted From Many Sources

Notes:

- EIU = Engine Interface Unit, DEU = Display Electronics Unit,
- CRT = Cathode Ray Tube, MDM = Multiplexer/Demultiplexer,
- C&W = Caution and Warning, GNC = Guidance Navigation & Control,
- DDU = Display Driver Unit, LPS = Launch Processing System,
- E = Effectors, S = Sensors

Figure 2.1-1. Space Shuttle Data Processing System Architecture



### **2.1.2 VEHICLE DERIVED**

- Full-time flight control augmentation is required dictating fly by wire placing the digital flight control computation system in the safety critical path.
- Engine actuator hardover commands are extremely critical requiring redundant summed inputs for voting to prevent erroneous commands.
- Data buses and remote power control devices are required to save weight.

)

)

)

## **2.2 DATA PROCESSING SYSTEM DESCRIPTION**

The DPS consists of the following key hardware and software:

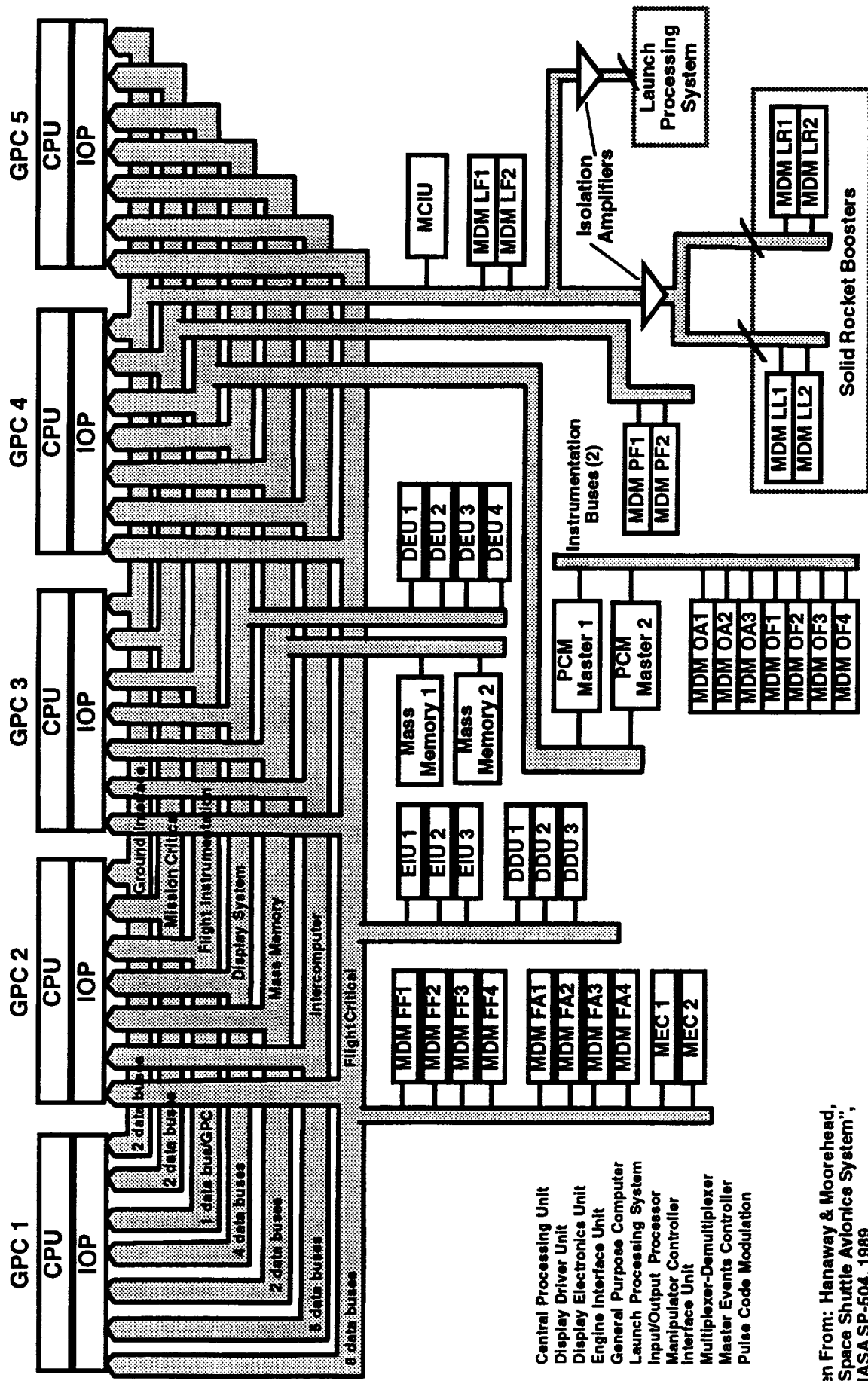
- Multiplexed data transmission with standardized subsystem interfaces over the 24 digital data buses
- 5 GPCs with interfaces interconnected by digital data buses into a parallel-redundant digital computation system
- Mass software program storage in two tape mass memory units
- Distributed I/O through remotely located multiplexer/demultiplexer (MDM) units over the digital data buses
- Communication with multifunctional displays and keyboards via the Display Electronics Unit (DEU)
- Time management using two Master Time Units.
- The PASS software.

### **2.2.1 DATA BUSES**

The use of GPC data buses is critical to operation of the shuttle DPS, and to safe and successful operation of the vehicle. The data bus architecture for the shuttle is shown in Figure 2.2-1. The shuttle data bus network consists of 24 twisted shielded wire pairs (data busses) which support the transfer of digital commands from the GPCs to vehicle hardware and the transfer of vehicle systems data to the GPCs. Each GPC has 24 serial digital data bus interface ports with functions allocated by criticality and use with no Hamming-type error protection. There are seven groups of busses. They consist of 8 flight critical (FC), 2 payload (PL), 2 launch/boost (LB), 2 Mass Memory Unit (MMU), 4 display keyboard (DK), 5 instrumentation/PCM master unit (IP), and 5 intercomputer communication (ICC) data buses.

### **2.2.2 GPC OPERATIONS**

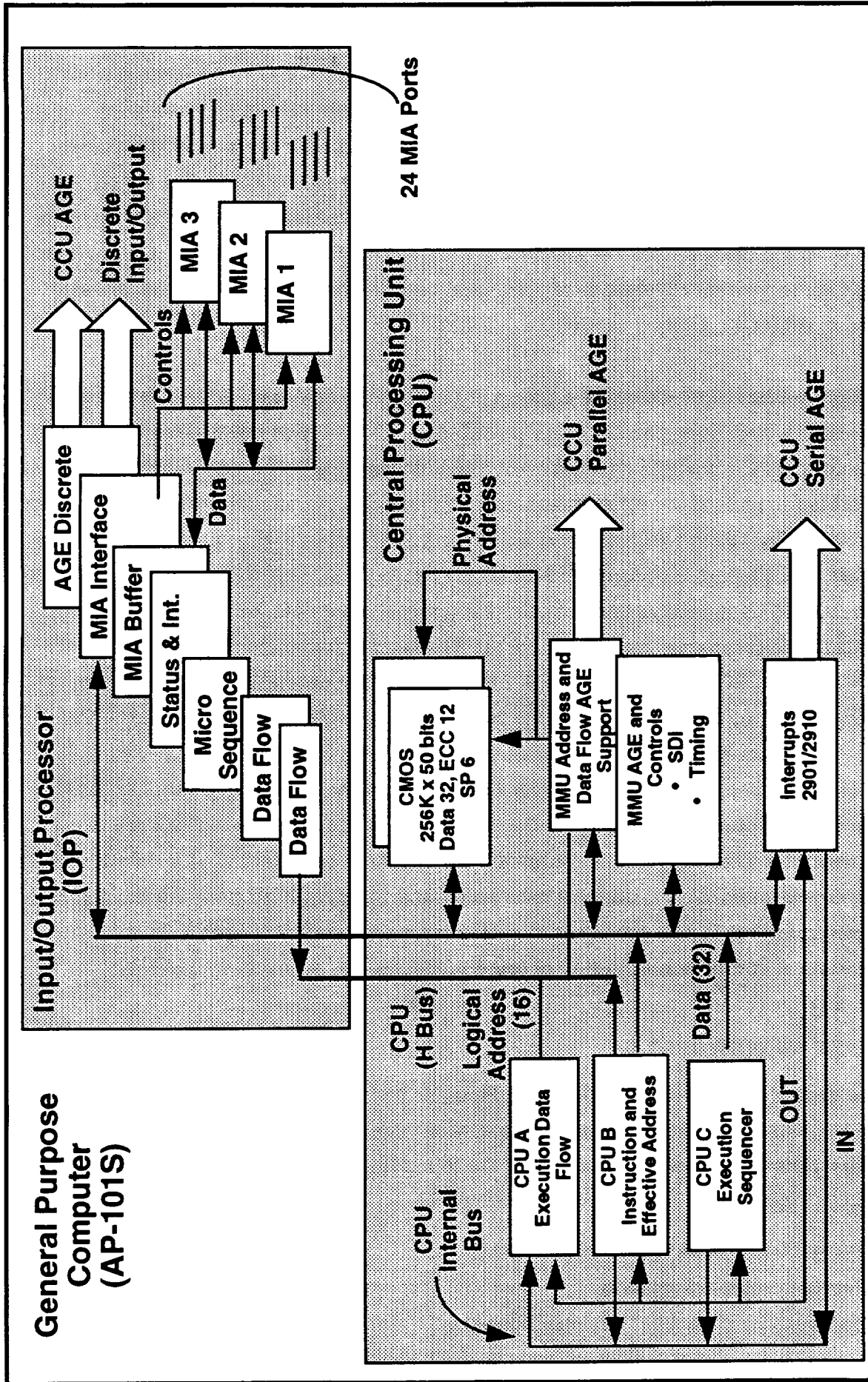
The GPC internal structure is depicted in Figure 2.2-2. As noted above, all 4 common GPCs operate in a redundant set. To prevent divergence while operating in this set, the synchronization method selected is to insert "sync points" at appropriate locations in the software. When a computer reaches one of these sync points it stops execution, notifies the



- CPU
- DDU
- DEU
- EIU
- GPC
- LPS
- IOP
- MCIU
- MDM
- MEC
- PCM
- Central Processing Unit
- Display Driver Unit
- Display Electronics Unit
- Engine Interface Unit
- General Purpose Computer
- Launch Processing System
- Input/Output Processor Interface Unit
- Manipulator Controller
- Multiplexer-Demultiplexer
- Master Events Controller
- Pulse Code Modulation

Taken From: Hanaway & Moorehead, "Space Shuttle Avionics System", NASA SP-504, 1989

Figure 2.2-1. Space Shuttle General Purpose Computer (AP-101S) Data Bus Architecture



Taken From: Buskirk, "AP-101S Space Shuttle GPC Design Handbook", IBM 85-C67-005, November 1985

Figure 2.2-2. Space Shuttle General Purpose Computer Internal Block Diagram

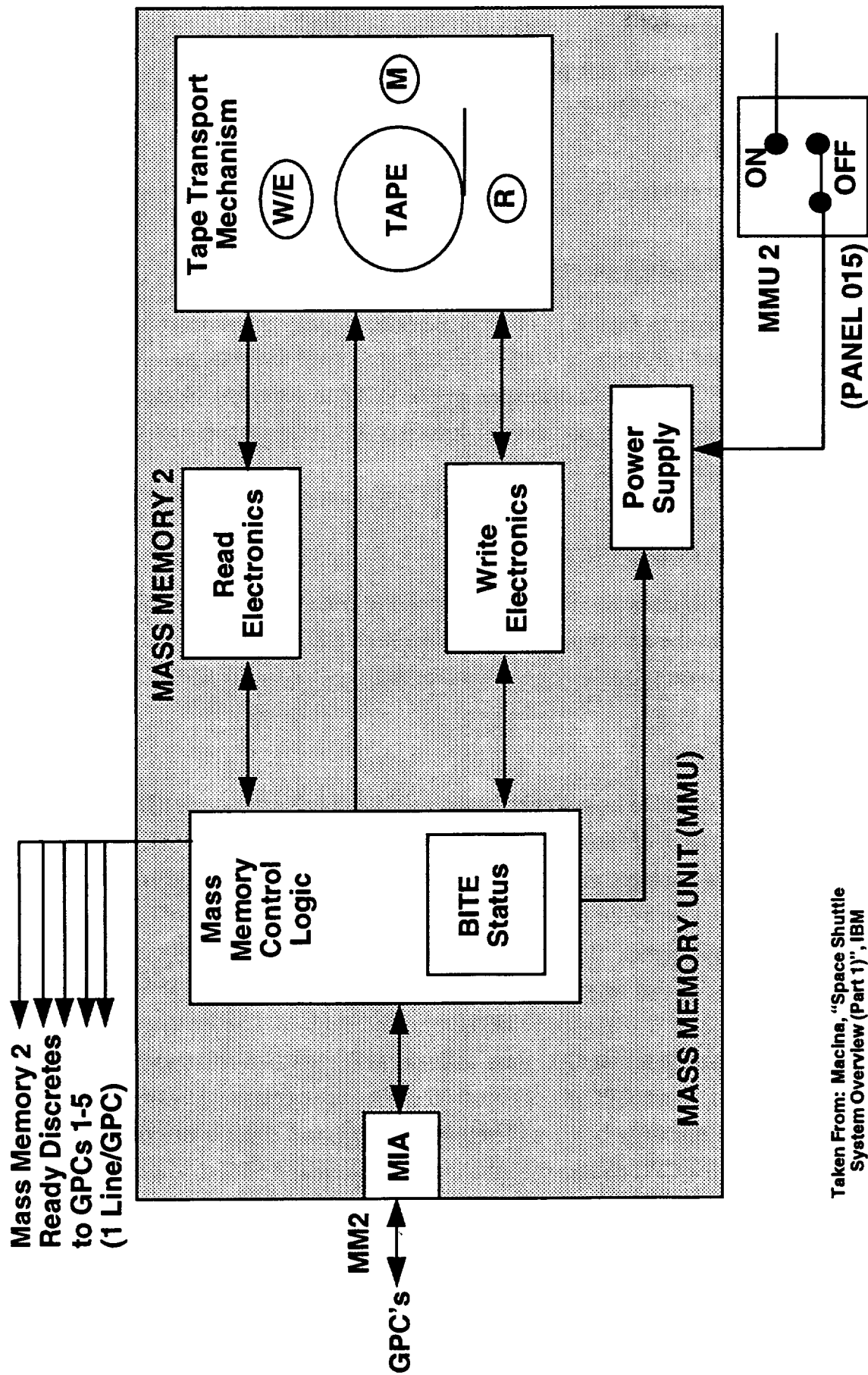
other computers by way of sync discrettes that it has reached that point and waits for receipt of corresponding sync discrettes from the rest of the redundant set before continuing processing. If all discrettes are not received within the preset time the synced computers resume execution and declare any non-responsive computer to be failed. The synch method is a software or soft sync approach as opposed to a hardware sync approach which utilizes a common clock to lock the processors in sync. In addition, all Shuttle GPCs receive the same data to prevent divergence of computed results since computed results are also compared to detect a computer failure.

For the flight critical input channels, a group of four buses, each of the four redundant GPCs controls one bus and listens on the other three buses. Control here means simply that only the controlling GPC transmits commands on the bus controlled. The transmitter for that bus is disabled in each of the other three GPCs so they can not transmit, but can only receive or listen on the bus. Each GPC will send a wakeup command over the bus it controls to each of the other GPCs before it transmits a request for sensor data. This wakeup command cues the listen only GPCs to receive and record the returned sensor data. Since the GPCs are synchronized, all computers request data from its sensor simultaneously with each of the other GPCs. Each GPC also controls one of four flight critical output buses. Critical outputs from each of the four GPCs are sent on the bus it controls to the effectors where the four inputs are voted providing only one output. Each GPC also controls one of the five ICC data buses. Once per cycle a summary word consisting of the sum of all critical outputs for that cycle is transmitted by each GPC to the other three GPC's in the redundant set. Each GPC compares its summary with the summary words of the other GPCs. If it does not agree with at least one other GPC it declares itself failed and removes itself from the set. There are also several other logical processes to determine a computer failure that have been implemented. A complete description of these processes can be found in [HAN89].

### **2.2.3 MASS MEMORY UNIT**

The mass memory unit (MMU) is shown in Figure 2.2-3. Two are installed in each orbiter. They are magnetic tape units with random access storage capacity of 4.2 million 32-bit words each. They provide nonvolatile storage for the following:

- System software
- Duplicate copies of application programs
- Overlay program segments
- Cathode-Ray Tube (CRT) display formats



Taken From: Macina, "Space Shuttle System Overview (Part 1)", IBM Presentation

Figure 2.2-3. Space Shuttle Mass Memory Unit Functional Block Diagram

- Prelaunch test routines
- Fault isolation diagnostic test programs
- I-loads (mission and hardware unique data)
- Checkpoint data
- Downlink data formats

#### **2.2.4 MULTIPLEXER/DEMULTIPLEXER UNIT**

The shuttle MDM is shown in Figure 2.2-4. It is a flexible multipurpose interfacing device. The MDM recognizes and responds to valid, correctly addressed data bus transmissions. Its sequence control unit (SCU) controls operation of the MDM in acquiring blocks of data. The 16 Input/Output (I/O) slots can be populated with a mix of 9 different types of analog, discrete, digital and special-purpose I/O modules.

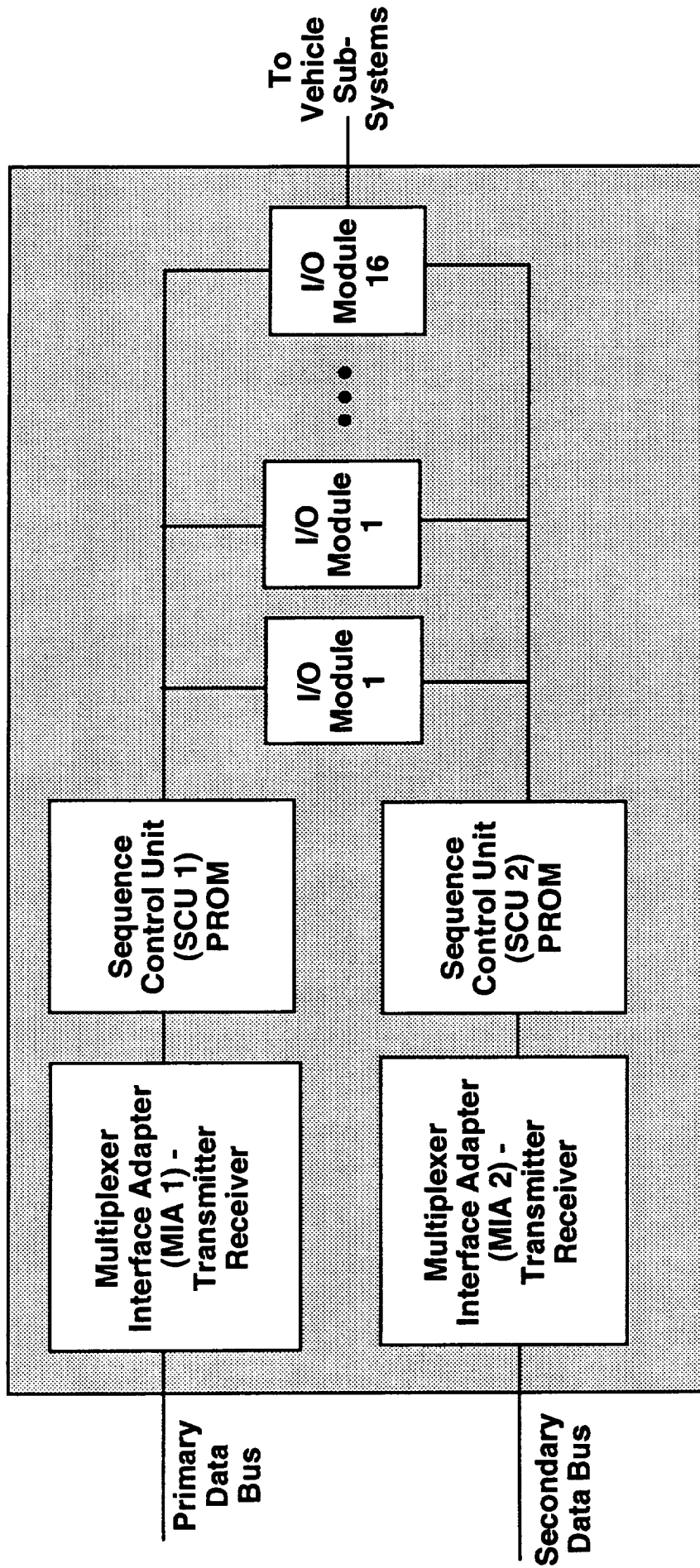
#### **2.2.5 DISPLAY ELECTRONIC UNIT**

The DEU is the hardware/software unit which drives the general purpose displays and accepts crew inputs from the alphanumeric keyboard. Each DEU has one digital data bus input and a special purpose processor with Random Access Memory (RAM).

#### **2.2.6 TIME MANAGEMENT**

The GPC uses a stable, accurate time source based on Greenwich mean time (GMT) for scheduling processing. Each of the five GPCs uses the Master Time Unit (MTU) to update its internal clock. There are three accumulators in each of the two MTUs on an orbiter. Each of the accumulators maintain both GMT and Mission Elapsed Time (MET), which can be updated by an external signal. Each accumulator is tied to a different flight critical forward MDM. Because of this arrangement, any one of the GPCs which is at least "listening" on strings 1,2, or 3 will receive MTU time and Built-In Test Equipment (BITE) status. Software compares internal GPC time with the MTU time sources and updates the internal clock as needed. Each GPC checks internal clock once per second against the MTU accumulator. If within tolerance ( $\leq 1$  millisecond), the internal clock is re-synchronized. If outside of tolerance, the GPC checks the other accumulators and GPC clocks until a within-tolerance time is found for updating. Procedures are available for resynchronizing out-of-tolerance clocks.





Taken From: Hanaway & Moorehead,  
 "Space Shuttle Avionics System",  
 NASA SP-504, 1989

Figure 2.2-4. Space Shuttle Multiplexer/Demultiplexer Block Diagram

)

.

.

)

.

.

)

## **2.3 SHUTTLE SOFTWARE ARCHITECTURE**

The architecture of the Space Shuttle software is shown in Figure 2.3-1. There are two sets of software resident in the GPCs: the system service software and the applications software. Key functions provided are summarized below.

### **2.3.1 APPLICATION PROCESSING SOFTWARE**

This is the application software to be executed to perform the activities needed to fly and operate the shuttle. The major categories of application processes are Guidance, Navigation and Control (GN&C), System Management and Vehicle Checkout. Only one major function at a time can operate in each GPC; however, different functions can operate in different GPCs in non-critical flight phases. GNC will usually operate in more than one GPC. Major functions are subdivided into mission-phase oriented blocks called operational sequences (OPS). Each OPS is associated with a specific memory configuration (MC) which can be loaded into a GPC from the MMU. Thus, all the software in a GPC at one time consists of the systems software and the OPS software (i.e., a MC). Transitions from one MC to another is called an OPS transition when the crew requests a new set of applications software to be loaded.

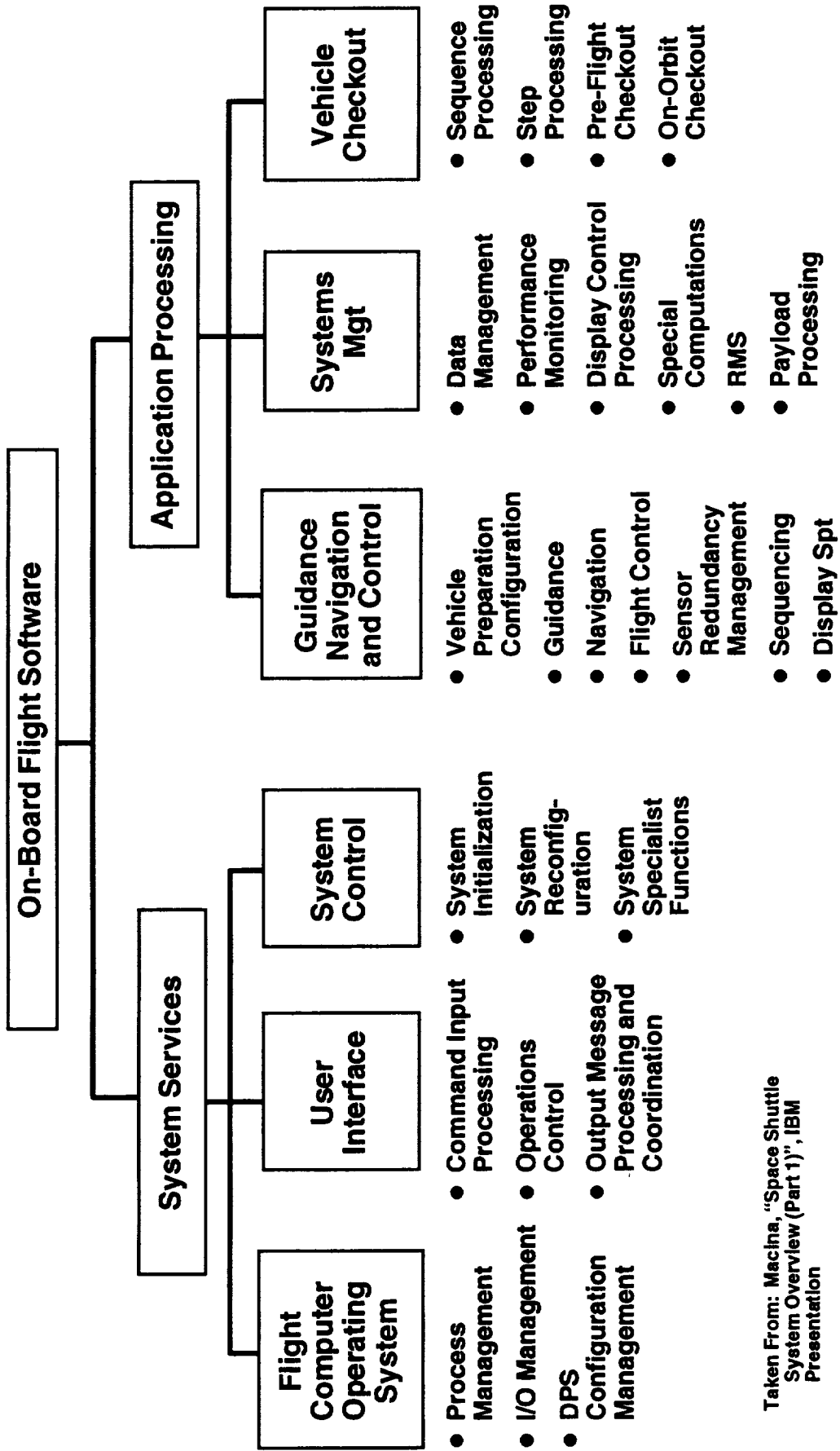
### **2.3.2 SYSTEM SERVICES SOFTWARE**

#### **2.3.2.1 Flight Computer Operating System Software Description**

The Flight Computer Operating System (FCOS) performs the same type functions as the SGOAA "Operating System" and consists of a synchronous foreground executive with a structure that uses an asynchronous priority driven background. The asynchronous priority driven background accommodates growth and the synchronous executive is predictable.

The FCOS kernel consists of the following three major functions.

- Process management: controls allocation of all internal computer resources.
- I/O Management: Controls allocation of I/O processing. This function performs all of the I/O functions including UIL, data bus management, and data base manager. It also includes the I/O error processing, special tests for other failure conditions and failure annunciation. I/O transactions are usually performed cyclically, except for those critical to vehicle safety, which are processed at a higher cyclical rate up to 25 Hz. Asynchronous requests from software are also handled.



Taken From: Macina, "Space Shuttle System Overview (Part 1)", IBM Presentation

Figure 2.3-1. Space Shuttle Primary Avionics Software Functional Structure

- **DPS Configuration Management:** Controls loading of computer memories and sequencing and control of the GPC and IOP operating states.

### **2.3.2.2 User Interface System Services Software Description**

This segment manages the sequencing of all processing and defines the associated CRT displays and keyboard options. These are the same type functions as the SGOAA I/O Data Services.

One of the key User Interface services is the Command Input Processing. This is the same type function as SGOAA I/O Data Services Data Acquisition. Command Input Processing includes crew inputs and the launch data bus used to communicate while on the ground.

Another important User Interface service is the output message processing and coordination software. This provides the same type functions as SGOAA I/O services data distribution.

### **2.3.2.3 System Control System Services Software Description**

This segment performs initialization and configuration control of the data processing complex including the data bus network. These are the same type functions as SGOAA Data System Management.

1

2

3

4

### 3. APPLICATION OF THE SGOAA TO THE SPACE SHUTTLE DPS

The SGOAA System Architecture is presented in Figure 3.1-1. This can be compared to the shuttle system architecture shown in Figure 1.1-1. A key difference in these architectures is the reliance (in the SGOAA) on the 6 classes of interfaces and use of standards in implementation. The SGOAA approach establishes independence between hardware and software entities at different interface levels to facilitate future change. The shuttle uses older methodologies and architectural concepts, with applications and services designed as monolithic or integrated entities which resist changes.

The SGOAA hardware and software architectures can be compared to and overlaid on the shuttle architectures to demonstrate the utility of the SGOAA in actual systems. Such comparisons and overlays are described below.

#### 3.1 HARDWARE ARCHITECTURE

The SGOAA Generic Processing External Hardware Architecture is shown in Figure 3.1-2. This can be compared to the shuttle hardware architecture shown in Figure 2.2-1. The hardware architecture diagrams are very similar. The following relationships of the various hardware units between the two systems can be established:

<u>SGOAA External Hardware Architecture</u>		<u>Space Shuttle Data Bus Architecture</u>
• GAP (S)	=	GPC (CPU/IOP), DEU, MMU
• LOCAL INTERCONNECT	=	MIA (DATA BUS)
• GAP (M)	=	MDM, PCMMU MEC, EIU, DDU
• SAP	=	MCIU

The General Avionics Processor (GAP) is a modular general purpose computer architecture as illustrated in Figure 3.2-1 and can be configured to all of the Space Shuttle Avionics requirements at the system architecture level. A GAP has one of two forms: one for standard general purpose use [GAP(S)] and one dedicated to the handling of multiplexing and demultiplexing I/O signals [GAP(M)]. A GAP(S) always includes a general purpose application processor modular function. A GAP(M) may contain a special purpose Programmable Read Only Memory (PROM) programmed processor, but its primary function is always the receipt, conditioning and distribution of I/O data. The Special Avionics Processor (SAP) is a special purpose processor designed to perform unique, not

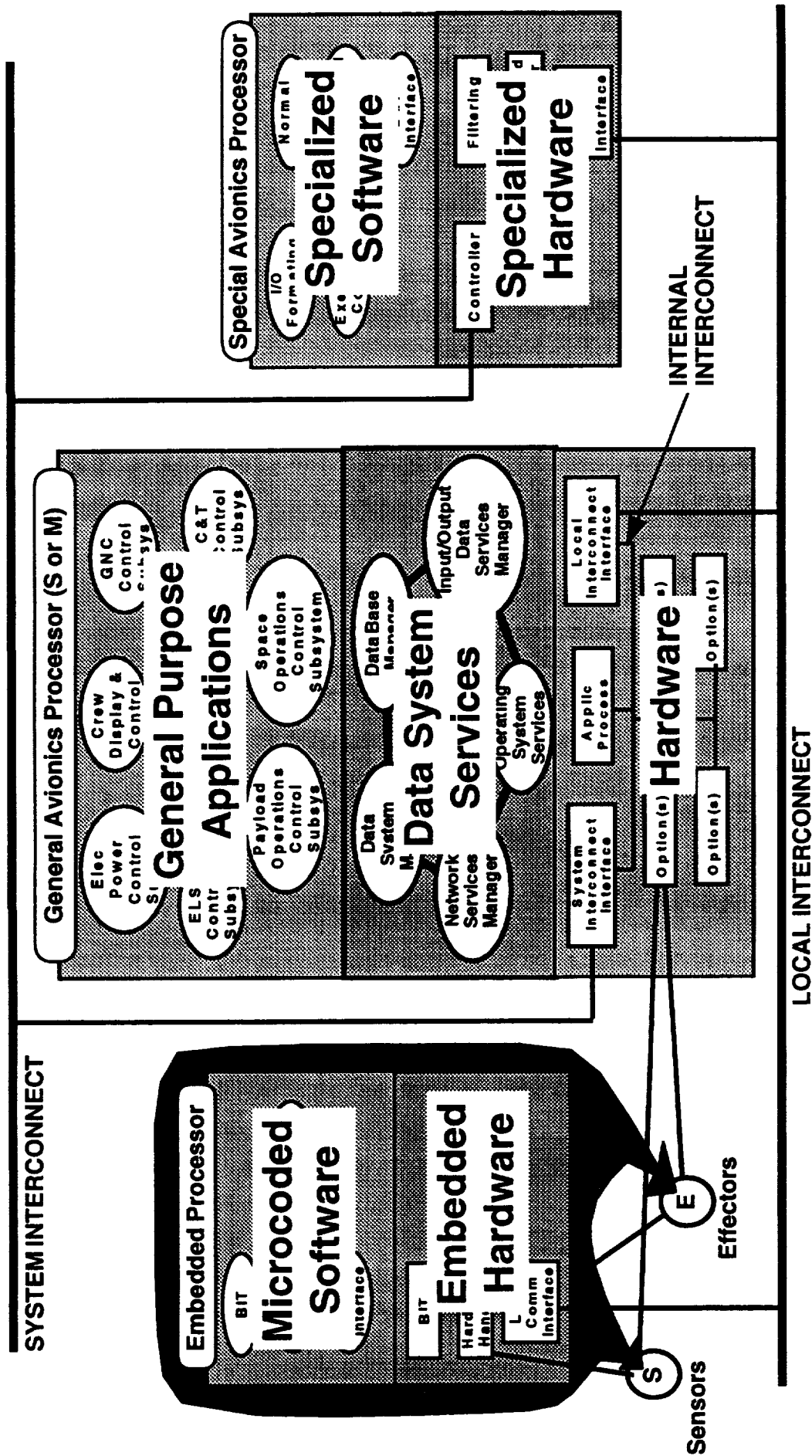


Figure 3.1-1. The Generic System Architecture Model Enables Flexible Use of Standards



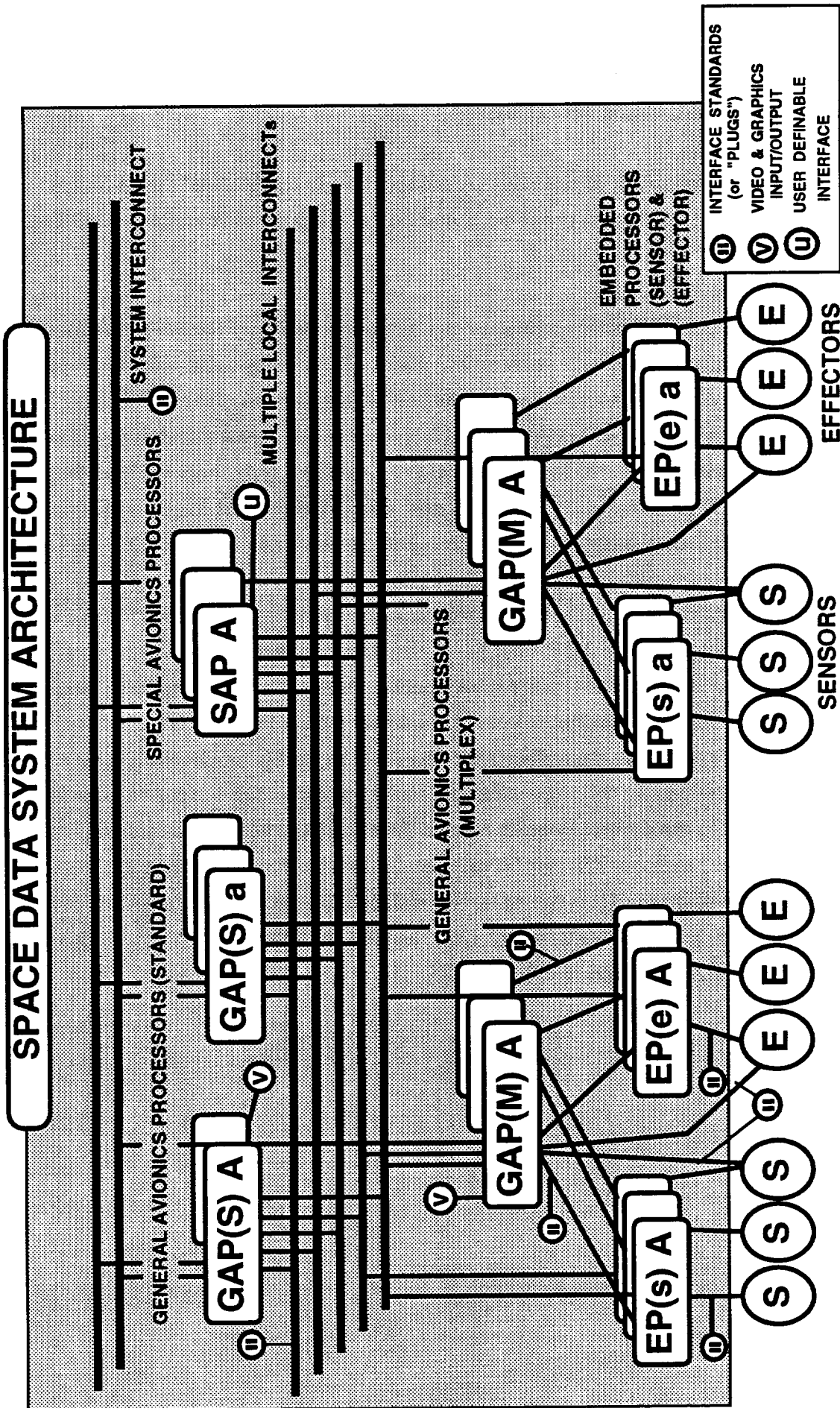


Figure 3.1-2 Generic Processing External Hardware Architecture Model and Interfaces Are Adaptable

general purpose processing functions and also does not have as its primary function the multiplexing and demultiplexing of I/O signals.

A description of the application of the SGOAA architecture to the internal Space Shuttle hardware architecture is contained in the Section 3.2 which discusses the internal hardware architecture.

The SGOAA and the Space Shuttle architectures function in basically the same manner. GAP(S) type units communicate with each other, with SAP type units and with GAP(M) type units over multiple local interconnects. The SGOAA system interconnect is not required for the Space Shuttle. GAP(M) type units perform intelligent multiplexing and demultiplexing of signals similar to the shuttle MDMs. GAP(M) type units such as the PCM Master Unit (PCMMU) communicate over dedicated local interconnect buses, Multiplexer Interface Adapter (MIA) data buses, to lower level EP type units such as the MDM OA1.

The Class 1 SGOAA hardware interfaces in Figure 3.1-2 also apply to the Space Shuttle data bus architecture shown in Figure 2.2-1. All hardware units communicate over a local interconnect bus. The SGOAA local interconnect path is required to be an accepted industry standard. The Space Shuttle multiplexer interface adapter (MIA) data bus is a modified 1553 standard bus. The video monitor interface from the DEU to the Space Shuttle Display Unit (DU) is another point where an interface standard could be applied. Interface standards could also be applied to the dedicated buses/signal lines from the MDM type units to the lower level Embedded Processor Effector (EP(e)) and Embedded Processor Sensor (EP(s)) units and to sensors/ effectors. User definable nonstandard interfaces from a SAP type unit, such as the Manipulator Controller Interface Unit (MCIU) to the manipulator, are also provided in the SGOAA architecture.

## 3.2 INTERNAL HARDWARE ARCHITECTURE

Figure 3.2-1 shows the SGOAA GAP internal architecture. As illustrated in this figure, the architecture is based upon an internal interconnect interface standard internal to the GAP and standard external interfaces from the modular GAP functions to external hardware entities. The interfaces are all SGOAA Class 1 hardware interfaces. The SGOAA interface classes are defined in Table 1. Figures 3.2-2 and 3.2-3 illustrates how seven of the Space Shuttle hardware units can be functionally built from the SGOAA GAP architecture. Figure 3.2-4 addresses two additional Space Shuttle Hardware units. Functions internal to SGOAA modules are not required to be standardized. Inputs and outputs to and from SGOAA modules are required to be standardized.

- **GPC** - This unit as shown in Figure 2.2-2 (AP-101S Block Diagram) contains three of the GAP(S) modular functions as shown in figure 3.2-2. They are the IOP equivalent to the GAP Local Interconnect Processing, the Central Processor Unit (CPU) equivalent to the GAP Application Processing and the Aerospace Ground Equipment (AGE) interfaces equivalent to the GAP Test and Checkout System Interface. When installed in the Shuttle, all test and checkout ground interfaces are by way of 2 dedicated data busses.
- **MMU** - This unit as shown in figure 2.2-3 contains three of the GAP(S) modular functions as shown in figure 3.2-3. They are the MIA equivalent to GAP Local Interconnect Processing, Mass Memory Control Logic equivalent to GAP Application Processing and the Read/Write Electronics and Tape Transport Mechanism equivalent to the GAP Auxiliary Memory Storage.
- **DEU** - This unit contains four of the modular GAP(S) functions as shown in figure 3.2-3. They are the MIA equivalent to GAP Local Interconnect Processing, processing of display data and formats in a special-purpose processor equivalent to GAP Application Processing, conversion of digital data to video/graphics form for display on the CRT equivalent to GAP Video/Graphics Processing, and interface to the keyboard equivalent to GAP I/O Processing.
- **MDM, MEC, EIU, DDU** - These four units all perform different tasks; however, each of these units contains only two of the modular GAP functions. These functions are MIA equivalent to GAP Local Interconnect Processing, and special processing of input and output data equivalent to GAP I/O Processing as shown in figures 3.2-2 and 3.2-3. Since the primary purpose of each of these four units is the handling of

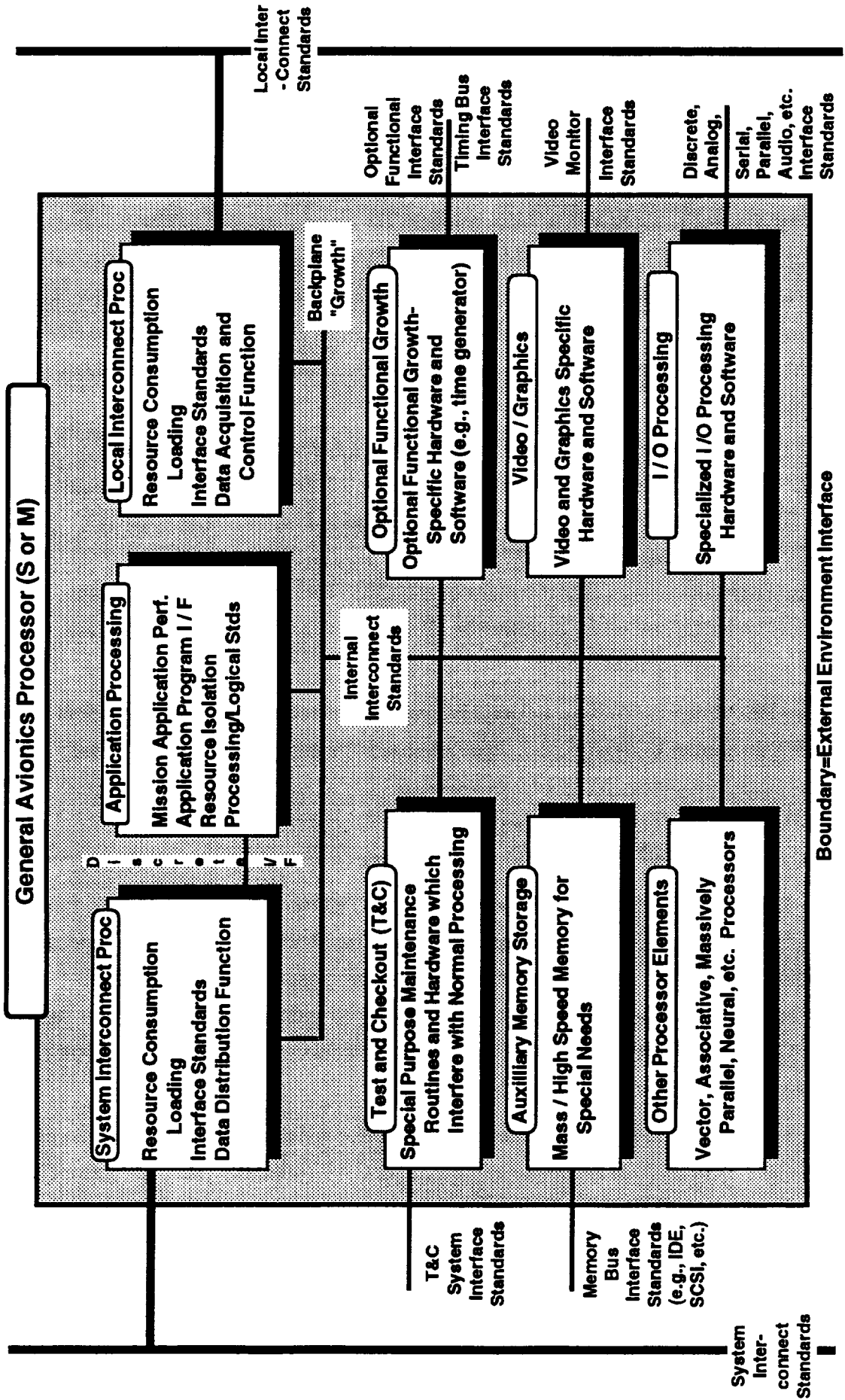
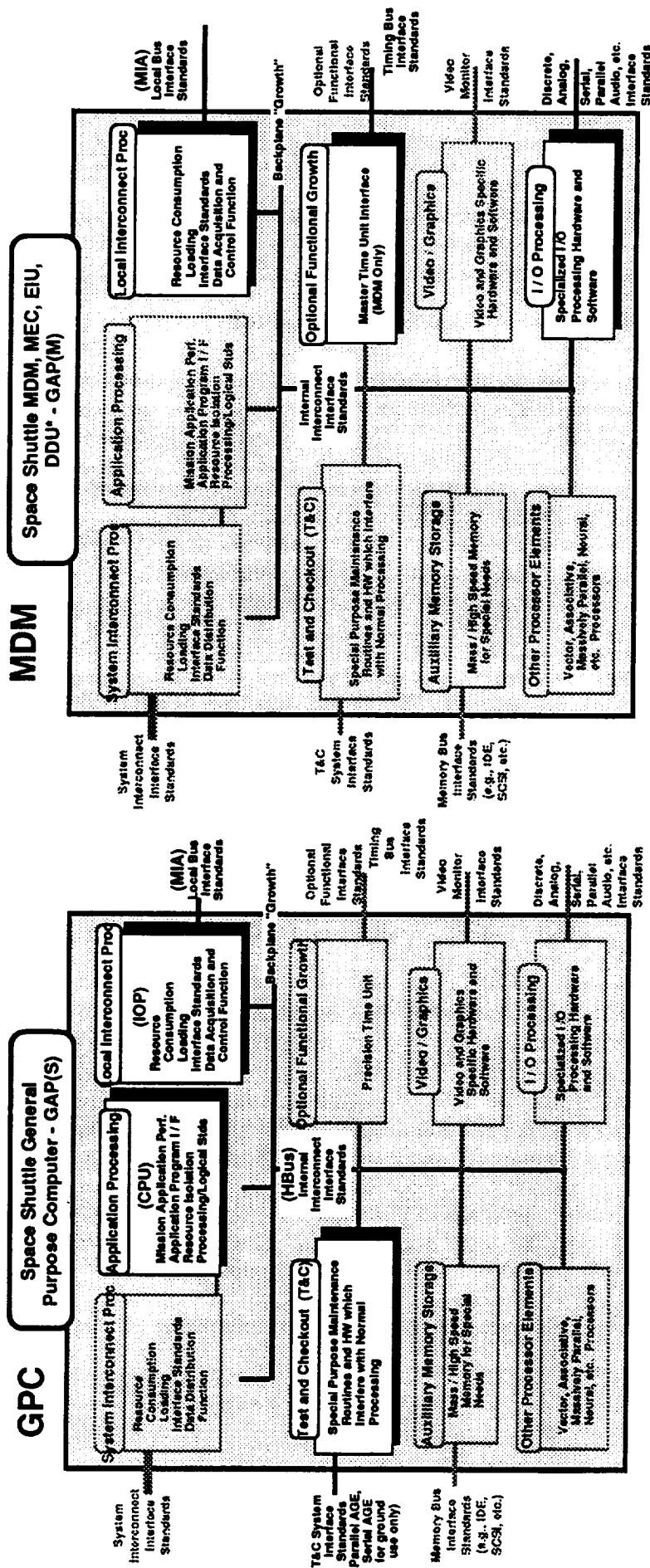


Figure 3.2-1. The Generic Processing Internal Hardware Architecture Can be Tailored

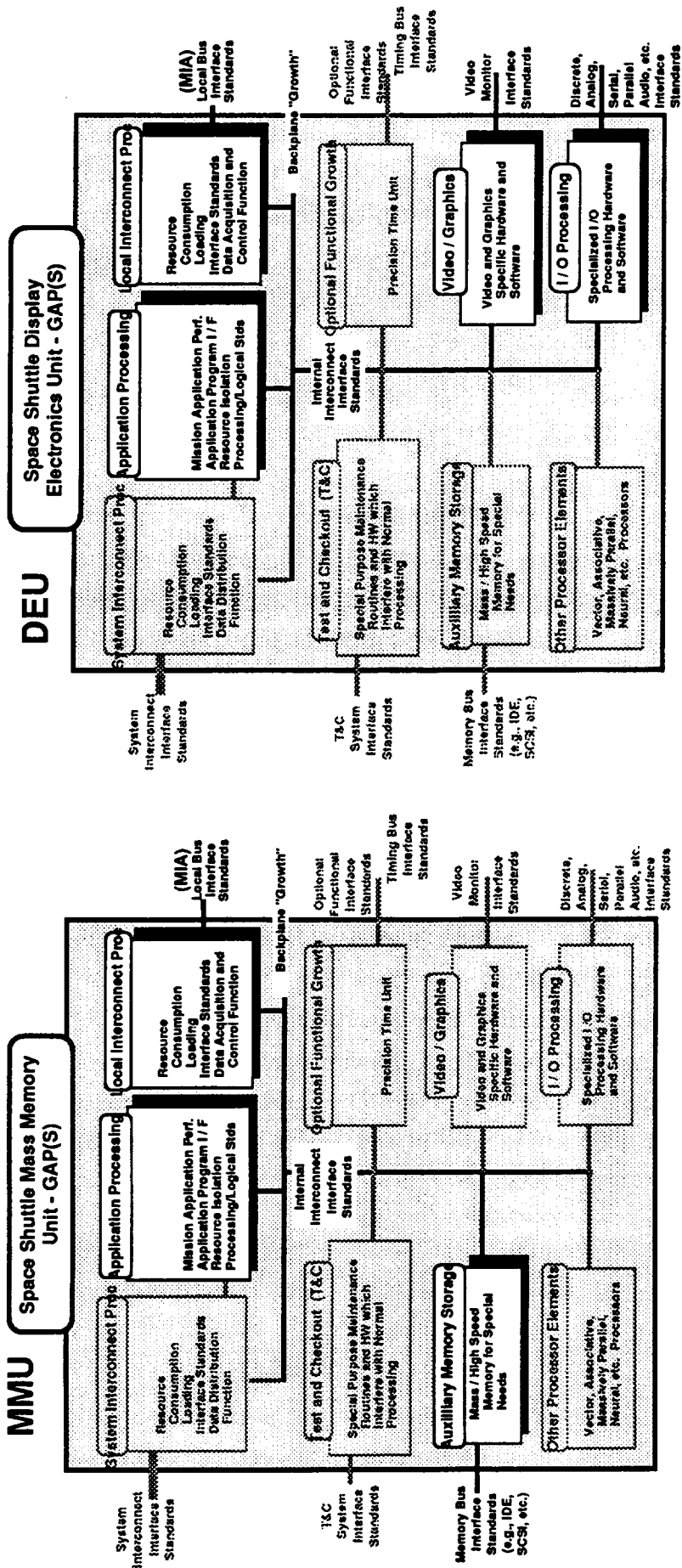
Table 3.2-1. Architectural Interface Classes

Class	Description
1	<p><b><u>Hardware-to-Hardware Direct:</u></b>            Class 1 hardware direct interfaces are the direct connections between different types of hardware such as needed to enable buses and communications links to address processors or needed to enable processors to address memory registers.</p>
2	<p><b><u>Hardware-to-Operating System Extension Software Direct:</u></b>            Class 2 hardware to operating system extension software direct interfaces are the direct connections between hardware registers and operating system extension service software or other software performing that function, such as drivers needed to enable address registers to move data packets from hardware to system service software, and service drivers which can respond to the data packets.</p>
3	<p><b><u>Operating System Services Software-to-Software (Local) Direct:</u></b>            Class 3 operating system service software to other software direct interfaces are the direct connections between operating system service code and other local software code sets, which enable operating system software to receive and interpret data packets, and pass them on to other software code which will process them locally.</p>
4	<p><b><u>Data System Services Software-to-Data System Services Software Logical:</u></b>            Class 4 system service software to other system service software logical interfaces are the indirect connections which enable local service software to determine the address of the intended software in other local or remote locations which need the register data being stored and to pass the data appropriately. Enables the handling of logical data transfers from source to user service</p>
5	<p><b><u>Data System Services Software-to-Applications Software Direct:</u></b>            Class 5 system service software to applications software direct interfaces are the direct connections which enable software service code to access and process data from local application software code.</p>
6	<p><b><u>Applications Software-to-Applications Software Logical:</u></b>            Class 6 applications software to applications software logical interfaces are the indirect connections which enable an application originating data to pass it to an application which needs to use the data, or enable an application needing data to determine the source from which the data must be obtained. These are logical data transfers from source to user. This interface provides the indirect connections that allow applications in different systems or in the same system to communicate, thus enabling applications software to interact across or within system boundaries to accomplish a mutual purpose. These interfaces may be applicable to applications executing in the same processor, in different processors in the same node or in different systems.</p>



\*Note: MDM = Multiplex-Demultiplexer, MEC = Master Events Controller, EIU = Engine Interface Unit, DDU = Display Driver Unit  
 Greyed out modules are those elements not applicable to the Shuttle architecture design  
 Text in modules may be seen full size in Figure 3.2-1.

Figure 3.2-2. The Generic Avionics Architecture Can be Applied to the Space Shuttle Processing Elements



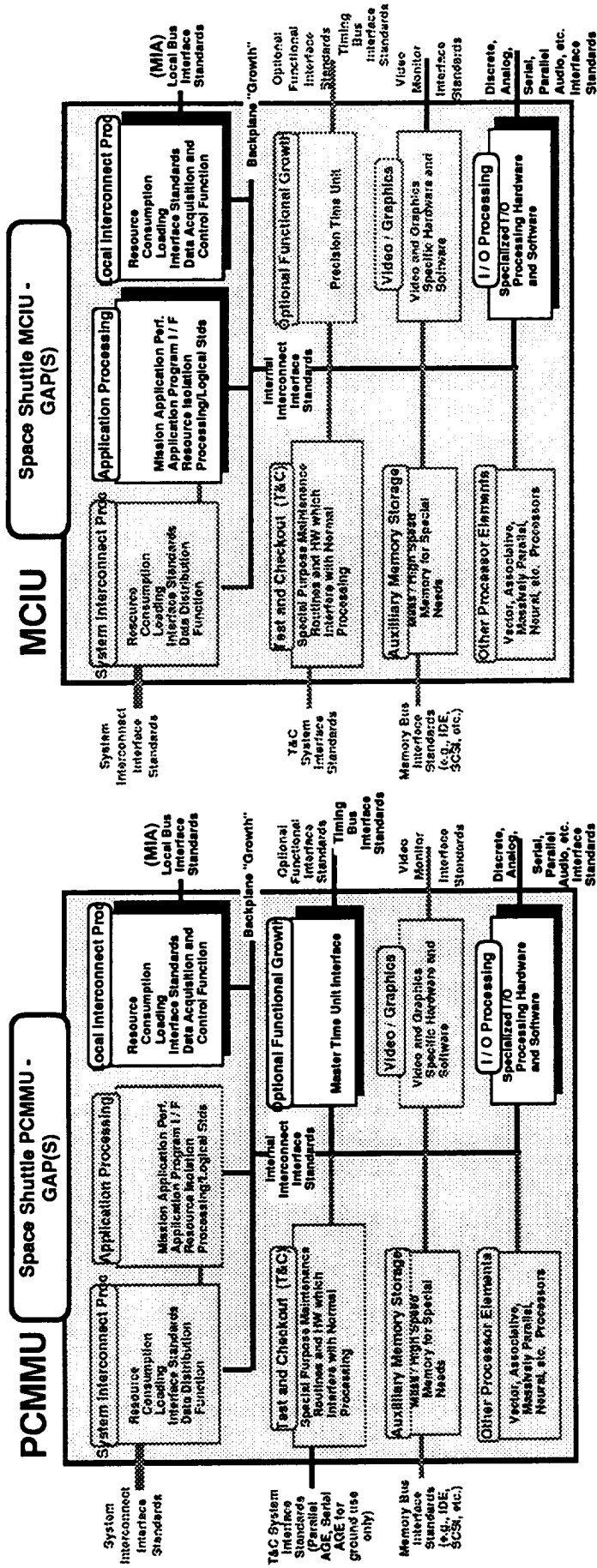
\*Note: MDM = Multiplex-Demultiplexer, MEC = Master Events Controller, EIU = Engine Interface Unit, DDU = Display Driver Unit  
 Grayed out modules are those elements not applicable to the Shuttle architecture design  
 Text in modules may be seen full size in Figure 3.2-1.

Figure 3.2-3. The Generic Avionics Architecture Can be Applied to the Space Shuttle Processing Support Elements

I/O data, they are designated as GAP(M) type units. The MDM was shown in figure 2.2-4. It can handle up to 9 different types of I/O processing and also contains a PROM programmed sequence control unit to control MDM operation. The Master Events Controller (MEC) has special control logic for processing of critical liftoff and stage-separation functions. The Engine Interface Unit (EIU) converts commands received from the GPCs over the data busses to engine bus protocol. The Display Driver Unit (DDU) converts the serial digital data stream received over the data bus to appropriate analog signals required to drive the various flight instruments.

- **PCMMU** - The PCMMU performs three of the modular GAP(M) functions as shown in Figure 3.2-4. The first is MIA communication over the data busses equivalent to GAP Local Interconnect Processing. The second is special I/O processing in which data received from the GPCs for insertion into the telemetry downlink data stream is formatted, commutated and configured. Additional special I/O processing is performed in the gathering of instrumentation data for use by the GPCs over dedicated instrumentation data buses from lower level MDMs. The third function is the Optional Functional Growth Interface to the MTU and distribution of the timing data over the data busses.
- **MCIU** - The MCIU is the control computer for the remote manipulator system (RMS). It has one data bus port used for receipt of moding and outer loop control signals from the GPCs. It is considered a SAP as its single function is to act as a control computer interface. The MCIU performs three of the modular GAP(M) functions as shown in Figure 3.2-4. The first is MIA communication over the single data bus equivalent to GAP Local Interconnect Processing. The second is application processing in which the moding and outer loop control signals from the GPCs are interpreted, RMS position data interpreted and the appropriate RMS digital commands created. The third function is special I/O processing in which the digital commands are sent and received from the RMS and analog RMS position data digitized.





\*Note: PCMMU = Pulse Code Modulation Master Unit, MCIU = Manipulator Controller Interface Unit  
 Greyed out modules are those elements not applicable to the Shuttle architecture design  
 Text in modules may be seen full size in Figure 3.2-1.

Figure 3.2-4. The Generic Avionics Architecture as Applied to Space Shuttle Other Processing Elements

)

.

)

)

### **3.3 SOFTWARE ARCHITECTURE**

The following requirements are extracted from paragraph 4.3.9 of the SGOAA Standard.

"An architecture prepared in accordance with this standard shall include requirements for data system services. This shall consist of at least requirements for input/output data services management, network services management, data base management, data system management, and an operating system." All systems will not require all services provided by the SGOAA data system services architecture. For these cases, the data system services architecture may be tailored to satisfy specific system requirements.

#### **3.3.1 SGOAA DATA SYSTEM SERVICES REQUIREMENTS**

The SGOAA data system services provides all the interfacing services needed by applications to operate and control the vehicle, and is comprised of the elements summarized below.

- Input/Output Data Services Management shall include at least requirements for Input/Output Data Services data acquisition, Input/Output Data Services data distribution and reports generation.
- Network Services Management shall include at least requirements for network services, network management, remote operation, network directory service, and network association control.
- Database Management shall include at least requirements for file services, distributed file transfer services, file transfer access and management, and node directory.
- Data System Management shall include at least requirements for configuration management, timing service control, initialization startup and reconfiguration, and health status and fault detection and recovery.
- Operating System shall include at least requirements for an Operating System (OS) kernel, a run time environment (RTE) and OS/RTE extensions.

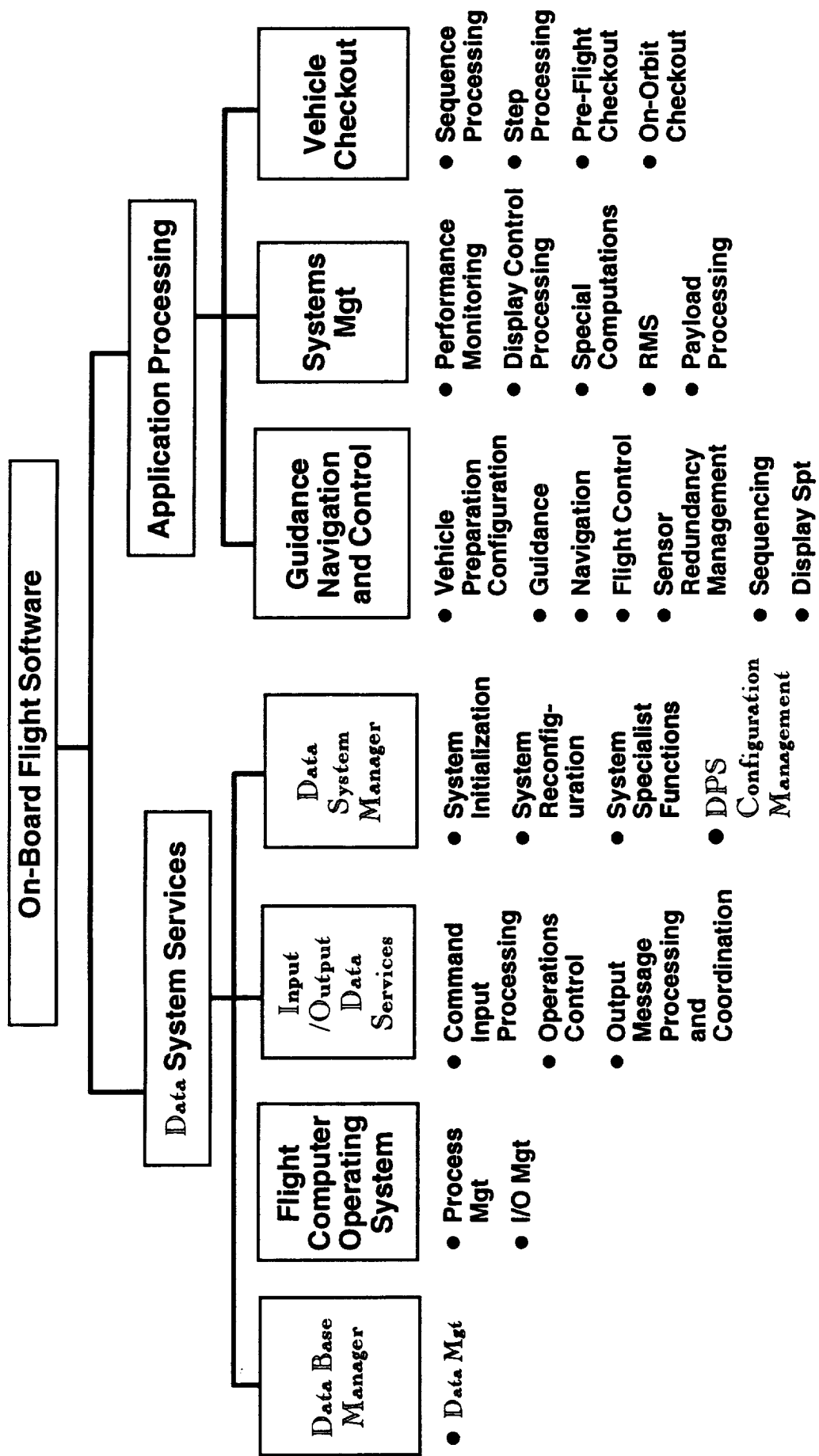
### **3.3.2 SPACE SHUTTLE ONBOARD FUNCTIONAL SOFTWARE ARCHITECTURE COMPARED TO THE SGOAA**

The Space Shuttle onboard functional software architecture structure was shown in figure 2.3-1. This architecture differs from the SGOAA "Operating System" architecture requirements in that the FCOS was developed expressly for the Space Shuttle, and does not satisfy open standards criteria as required by the SGOAA. In addition, in order to satisfy SGOAA requirements, the Space Shuttle architecture would have to be modified at a functional level as shown in Figure 3.3-1. The primary changes are:

- Change "System Services" name to "Data System Services"
- Change "User Interface" name to "Input/Output Data Services Management"
- Change "System Control" name to "Data System Manager"
- Add a second level entity called "Data Base Manager"
- Move "DPS Configuration Management" from FCOS to be a subset of "Data System Manager"
- From "Application Processing Systems Management" move "Data Management" to be a subset of the "Data System Services Data Base Manager"

Figure 3.3-2 shows how the SGOAA Space Data System Services would be tailored to provide the Space Shuttle software system services. Shaded areas are SGOAA data system services not required by the Space Shuttle DPS. The SGOAA Network Services Manager is not required as all Space Shuttle DPS communication is over the 24 data buses and not via a system interconnect network. The other deleted elements are self explanatory.

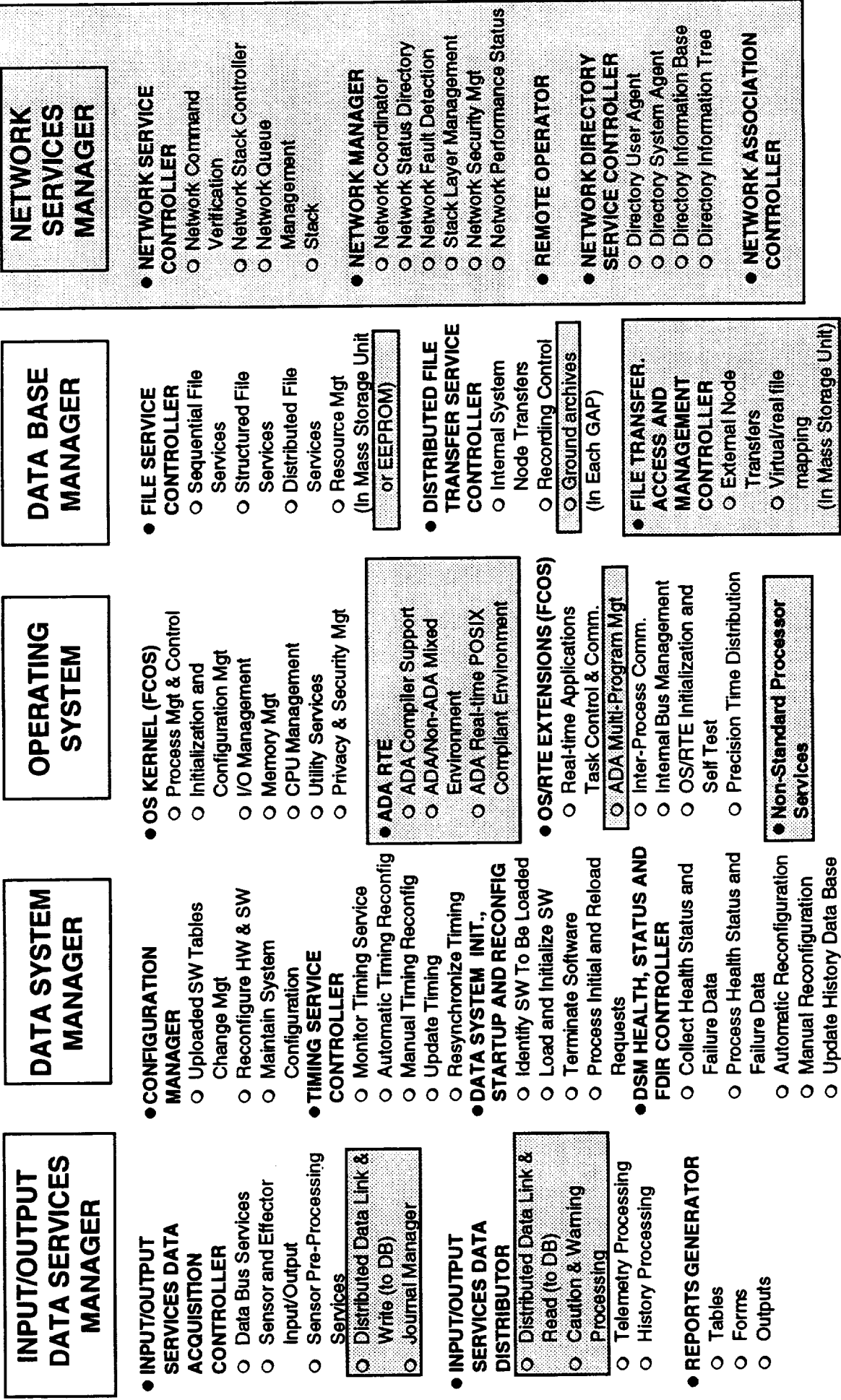
The Space Shuttle FCOS presently performs all SGOAA OS Kernel and OS/RTE functions. To be compliant with the SGOAA, the FCOS would be upgraded to meet open standards criteria or a waiver would be obtained to allow continued, unmodified FCOS usage.



3.3.3

Open Face Font shows changes from standard Shuttle software structure

Figure 3.3-1. Space Shuttle Primary Avionics Software If Structured Based on the SGOAA Model



☐ = Elements Deleted

Figure 3.3-2. Space Shuttle Data Processing System Functions Can Be Tailored from the SDSS

As in the SGOAA, applications processing is a function of user needs. To comply with the SGOAA requirements, the Space Shuttle application software would have to be modified to comply with the requirements of the SGOAA Class 5 and 6 interfaces as defined in Table 3.2-1. A key requirement of the SGOAA is that direct application to application communication is not allowed. All application to application software communications shall be implemented by use of system services. All communication must be through a Class 5 direct standard interface to system services to provide the direct communications path between applications. An estimate of the extent of the modifications that would be required is outside the scope of this paper.

)

.

.

)

.

.

)



#### **4. CONCLUSIONS**

As was shown above, the SGOAA was found to be capable of satisfying the functional architecture requirements of the Space Shuttle DPS. The SGOAA GAP internal and external architectures can be tailored to satisfy the Space Shuttle DPS hardware architecture requirements. In order to be compliant with the SGOAA, accepted industry standards are required to be used at all interface points or waivers be obtained. The Space Shuttle does not satisfy this requirement since it was designed at a very early stage in the development of standard interfaces. Were it to be designed today the requirement for the use of standard interfaces could be met. The SGOAA Space Data Systems Services architecture was also shown to be capable of being tailored to satisfy the Space Shuttle requirements.

)

)

)



**DISTRIBUTION LIST FOR LESC-30753  
IMPLEMENTING THE SPACE SHUTTLE DATA  
PROCESSING SYSTEM WITH THE SPACE  
GENERIC OPEN AVIONICS ARCHITECTURE - CONCLUDED**

**NAVAL AIR WARFARE CENTER**, AIRCRAFT DIVISION,  
WARMINSTER, PA 18974-0591  
RICHARD J. PARISEAU/CODE 102A  
RICHARD S. MEJZAK/CODE 2021

**MITRE CORPORATION**, 202 BURLINGTON ROAD, BEDFORD, MA 01730-1420  
JACK SHAY/DIRECTOR OF SYSTEMS DEVELOPMENT

**MR. AL ANDERMAN**, (ROCKWELL), 930 3rd STREET, APT. 302,  
SANTA MONICA, CA 90403

**WESTAR CORP.**, 6808 ACADEMY PKWY EAST, NE, BLDG C, SUITE 3,  
ALBURQUERQUE, NM 87109  
CHRIS DE LONG

**BOEING CORP.**, P.O. BOX 3999, SEATTLE, WA 98124-2499  
RICHARD FLANAGAN  
AL COSGROVE

**COMPUTING DEVICES INTL**, 8800 QUEEN AVENUE SOUTH,  
BLOOMINGTON, MN 55431  
JIM JAMES, M/S BLCS1D  
DOCK ALLEN, M/S BLCW2S

**TEXAS INSTRUMENTS**, 6550 CHASE OAKS BLVD, PO BOX 869305,  
PLANO, TX 75086  
DR. CHUCK ROARK/MS 8481

**MCDONNELL DOUGLAS CORPORATION**, 1801 E. ST ANDREW PLACE,  
SANTA ANA, CA 92705  
TERRY RASSET/MS A208

**HUGHES AIRCRAFT**, P. O. BOX 92426, LOS ANGELES, CA 90009-2426  
JOHN GRIFFITH, RE/RI/B500

**FAIRCHILD SPACE**, 20301 CENTURY BLVD., GERMANTOWN, MD. 20874  
JOHN SCHNEIDER, FLIGHT DATA SYSTEMS

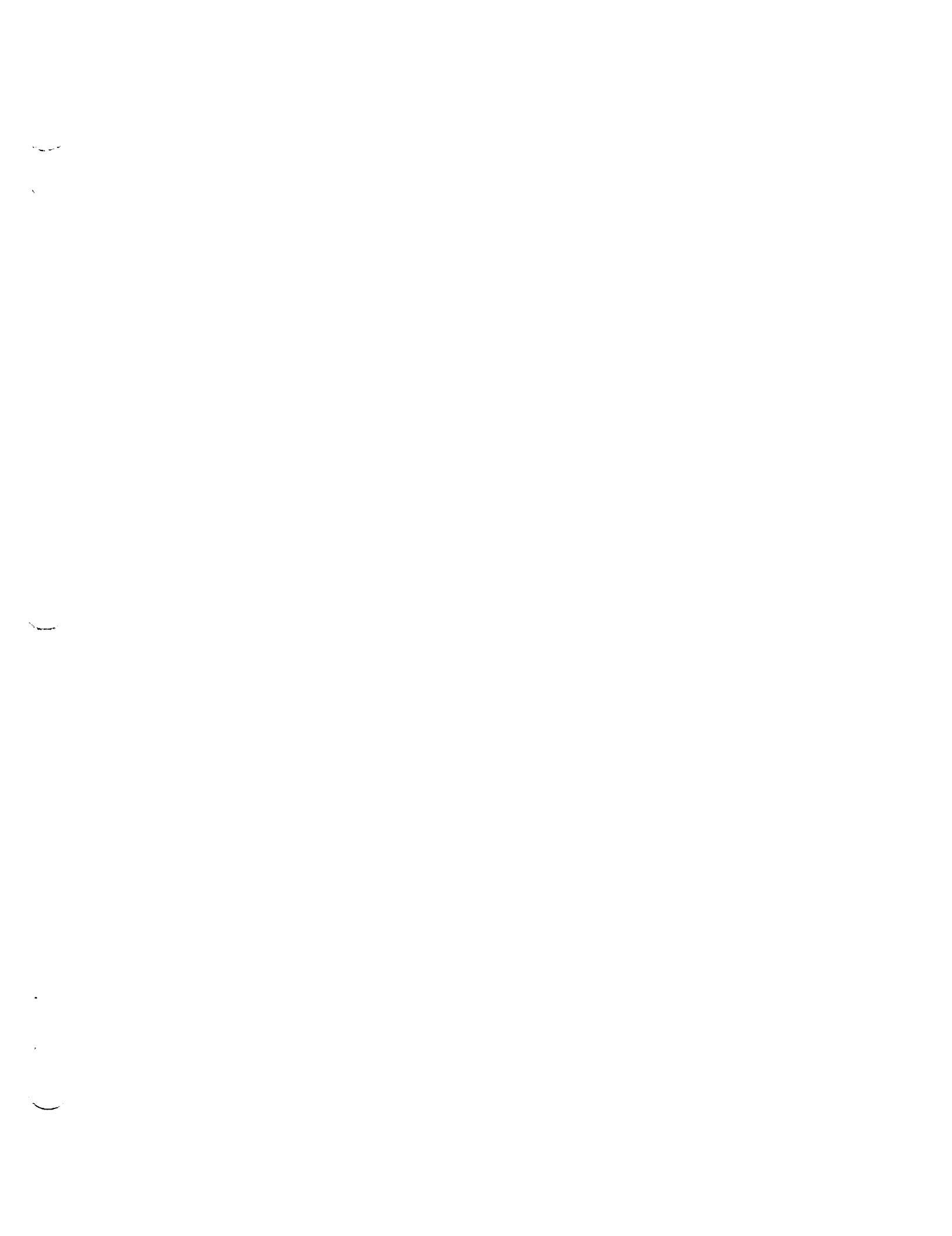
## REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE July 1993	3. REPORT TYPE AND DATES COVERED Contractor Report	
4. TITLE AND SUBTITLE Implementing the Space Shuttle Data Processing System with the Space Generic Open Avionics Architecture			5. FUNDING NUMBERS	
6. AUTHOR(S) Richard B. Wray				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Lockheed Engineering and Sciences Company 2400 NASA Road 1 Houston, Texas 77573-3799			8. PERFORMING ORGANIZATION REPORT NUMBER LESC-30753	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Lyndon B. Johnson Space Center Flight Data Systems Division, EK Houston, Texas 77058			10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA CR-188254	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified/unlimited Available from National Technical Information Service 5285 Port Royal Road, Springfield, VA 22161, (703) 487-4600 Subject Category 66			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper presents an overview of the application of the Space Generic Open Avionics Architecture (SGOAA) to the Space Shuttle Data Processing System (DPS) architecture design. this application has been performed to validate the SGOAA and its potential use in flight critical systems. The paper summarizes key elements of the Space Shuttle avionics architecture, data processing system requirements and software architecture as currently implemented. It then summarizes the SGOAA architecture and describes a tailoring of the SGOAA to the Space Shuttle.  The SGOAA consists of a generic system architecture for the entities in spacecraft avionics, a generic processing external and internal hardware architecture, a six class model of interfaces and functional subsystem architectures for data services and operations control capabilities. It has been proposed as an avionics architecture standard with the Nation Aeronautics and Space Administration (NASA), through its Strategic Avionics Technology Working Group, and is being considered by the Society of Aeronautic Engineers (SAE) as an SAE Avionics Standard. This architecture was developed for the Flight Data Systems Division of JSC by the Lockheed Engineering and Sciences Company, Houston, Texas.				
14. SUBJECT TERMS avionics, space shuttle, generic avionics, architecture, data processing system, interfaces			15. NUMBER OF PAGES	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT unclassified	20. LIMITATION OF ABSTRACT UL	





**DISTRIBUTION LIST FOR LESC-30753  
IMPLEMENTING THE SPACE SHUTTLE DATA  
PROCESSING SYSTEM WITH THE SPACE  
GENERIC OPEN AVIONICS ARCHITECTURE**

**NASA**

EK111/D. M. PRUETT (10)  
PT4/E. M. FRIDGE (5)  
AMES/E. S. CHEVERS  
EK3/J. BELL (2)  
EK3/D. JIH

EG1/D. P. BROWN  
EG111/K. J. COX  
JM-2/S. McDONALD (30)  
EK3/R. S. DAVIS

**LESC**

C18/J. R. THRASHER  
C18/E. A. STREET  
C18/R. E. SCHINDELER  
C18/J. STOVALL (10)  
B11/G. J. MOORMAN  
C106/P. G. O'NEIL

C18/G. L. CLOUETTE  
C18/R. W. WRAY (10)  
C18/M. W. WALRATH  
C18/B. L. DOECKEL  
C29/P. HOPKINS

C87/M. W. BRADWAY  
(FOR SATWG REDISTRIBUTION)

C18/JEAN FOWLER  
(MASTER + 2 COPIES)

B15/LESC LIBRARY (2)

**LESC**, 144 RESEARCH DRIVE, HAMPTON, VA 23666  
ROY WENDE  
PHIL MARTIN

**SAE/ASD**, SAE INTERNATIONAL, 400 COMMONWEALTH AVE, WARRENDALE,  
PA. 15096  
BARBARA ROTH (FILE) RICH VANDAME

**MITRE CORPORATION**, 1120 NASA ROAD 1, HOUSTON, TX 77058  
STEVE BAYER (2)

**UHCL**, UNIVERSITY OF HOUSTON - CLEAR LAKE, 2700 BAY AREA BLVD.  
BOX 444, HOUSTON, TEXAS 77058  
CHARLES HARDWICH

**NIST/CSL**, FRITZ SCHULTZ, BLDG 225, ROOM B266,  
GATHISBURG, MD 20899

**LASC**, 86 S. COBB ST., MARIETTA, GA 30063  
RICK HARWELL, D/73-D2, ZONE 0685