

F. Byrne  
G. V. Doolittle  
R. W. Hockenberger

## Launch Processing System

**Abstract:** This paper presents a functional description of the Launch Processing System, which provides automatic ground checkout and control of the Space Shuttle launch site and airborne systems, with emphasis placed on the Checkout, Control, and Monitor Subsystem. Hardware and software modular design concepts for the distributed computer system are reviewed relative to performing system tests, launch operations control, and status monitoring during ground operations. The communication network design, which uses a Common Data Buffer interface to all computers to allow computer-to-computer communication, is discussed in detail.

### Introduction

The Space Shuttle program is revolutionary in that, for the first time, space flight hardware will be reusable. Additionally, Shuttle ground operations will be streamlined to facilitate ground turn-around and in-orbit support with minimum crews. Specifically, the following program guidelines prove to be the most demanding factors in the design of Ground Support Equipment for Space Shuttle checkout [1]: 1) launch rate of 40 per year with a National Aeronautics and Space Administration (NASA) fleet size of three vehicles and 2) a demanding schedule of 160 hours for processing time from landing to launch with a minimum of manpower for vehicle checkout.

NASA, at the John F. Kennedy Space Center in Florida, is designing and acquiring a Launch Processing System [2], an important part of the Ground Support Equipment (GSE), to support launch site operations more efficiently than on previous programs. The Launch Processing System (LPS) will provide automatic control of GSE and Shuttle systems for test and operations, real-time data analysis and information display, and efficient recall of test data and engineering files to support Shuttle ground operations. Modern automation techniques, off-the-shelf components, and modular design are being used to achieve these goals. Significant technical challenges with today's state-of-the-art technology must be met. The LPS will be implemented in the next few years for operation in the 1980s.

IBM has been under contract to the National Aeronautics and Space Administration since May 1974 (contract completion in March 1979) and is responsible for

requirements analysis, allocation of functions among the system elements, and the total software development process.

Program guidelines, when evaluated with the Shuttle operations philosophy, lead to an LPS design with the following characteristics:

- High degree of automation
- Standard and modular hardware and software
- General purpose, multiple use, high density, non-dedicated consoles
- Test-engineer-oriented language for application programming
- Readily available planning and engineering information

### Frequently used abbreviations

CCMS	Checkout, Control, and Monitor Subsystem
CDBFR	Common Data Buffer
CDS	Central Data Subsystem
FEP	Front End Processor
FIFO	First-in, first-out
GOAL	Ground Operations Aerospace Language
GPC	General Purpose Computer
GSE	Ground Support Equipment
HIM	Hardware Interface Module
LDB	Launch Data Bus
LPS	Launch Processing System
PCM	Pulse Code Modulation
PDR	Processed Data Recorder
SPA	Shared Peripheral Area

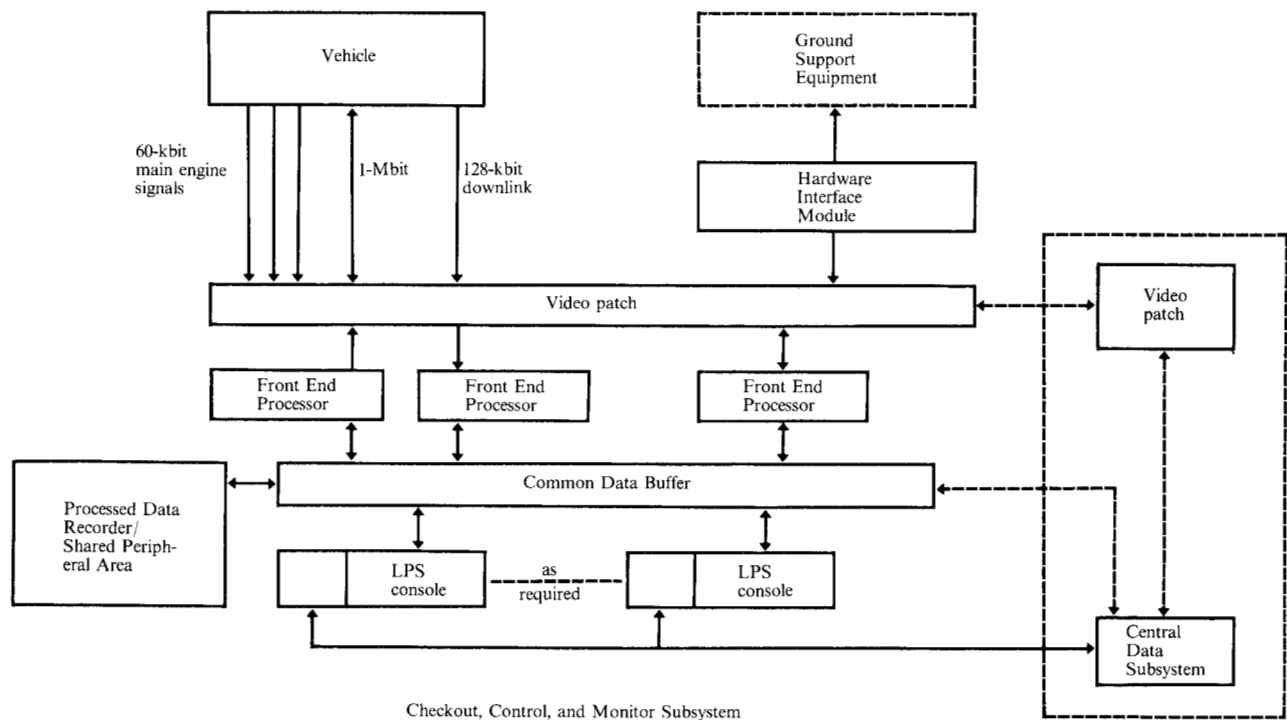


Figure 1 Launch Processing System configuration.

### System description

The Launch Processing System is an integrated system consisting of a Checkout, Control, and Monitor Subsystem (CCMS) and a Central Data Subsystem (CDS); see Fig. 1. These two subsystems share in the management of test and launch operations for the Shuttle.

The Central Data Subsystem consists of two large computers with common mass memory and communication processors to interface with the CCMS for real-time data transfer and to support remote terminal access. In general, the CDS performs the following four tasks:

1. Records selected test data in real time and allows rapid access to it from the CCMS and remote consoles.
2. Supports the CCMS software development and test by providing an interactive simulation capability for representing the GSE and the Shuttle.
3. Provides a facility for development, control, and maintenance of all CCMS software. It incorporates compilers, assemblers, and data bases for CCMS.
4. Provides a capability to maintain and access an engineering data file.

The CCMS consists of computers, displays, controls, data transmission devices, and electronic interface devices which control and monitor the Shuttle vehicle and Ground Support Equipment throughout the operations at the launch site. According to this design, operations

in the Orbiter Processing Facility, in the high-bay area of the Vertical Assembly Building, and at the pad are controlled from one of two firing rooms in the Launch Control Center. The CCMS hardware in stand-alone configurations is used for hypergolic module test and requalification; bench maintenance equipment tests; and possibly for solid rocket booster, external tank, and payload checkout. A functional representation of the CCMS launch support configuration is included in Fig. 1.

#### • Console subsystem

The checkout and monitor functions are performed in the console computers. The console subsystem provides the primary man-machine interface for all LPS operations. Checkout processing consists of the following four functions: 1) test procedure execution, 2) test article control, 3) status monitoring, and 4) LPS functional test.

#### • Common Data Buffer

The Common Data Buffer (CDBFR) is the communication center of the CCMS. The buffer provides the means for computer-to-computer communication. It maintains measurement and status information of the vehicle, GSE, and LPS systems, and contains time information (for example, Greenwich Mean Time) used as references by LPS subsystems.

- *Front End Processor*

The Front End Processors (FEPs) provide all communication with the Ground Support Equipment and vehicle. The FEPs perform all data pre-processing for console computers to support uplink and downlink measurement requests. The functions of the Front End Processors are to

1. Update all measurements associated with GSE, vehicle, and LPS subsystems in the CDBFR at requested sample rates.
2. Provide uplink services for checkout programs.
3. Provide for exception monitoring of specified measurements.
4. Provide for vehicle-to-ground communication.
5. Provide for on-line measurement parameter change processing.

- *Processed Data Recorder/ Shared Peripheral Area*

The primary function of the Processed Data Recorder (PDR) computer is to selectively record data written into the Common Data Buffer. The PDR records, as a minimum, all measurement change activity, computer-to-computer communication parameters, CCMS error conditions, and selected software parameters from test procedures being executed in each console computer. The PDR is configured to handle, as a minimum, approximately 50000 16-bit words per second.

The primary function of the Shared Peripheral Area (SPA) is to support the post-processing of data logged on the PDR. The Shared Peripheral Area serves as a backup for the Processed Data Recorder, if that system fails, by switching to the processing function.

### **Functional description**

The CCMS hardware and software are designed to minimize maintenance requirements during the operational phase. Both are structured in modules, performing specific functions with standard interfaces between modules and subsystems.

- *Software*

A common operating system nucleus exists in every computer in the CCMS configuration. A comprehensive software system build process is provided by the CDS to allow constructing unique software configurations by a predefined set of procedures to satisfy various LPS, vehicle, and GSE configurations without modifying CCMS software modules.

- *Hardware*

Hardware Interface Modules (HIMs) at specific GSE locations issue control signals and monitor measure-

ments (see Fig. 1). They are connected through data buses and a video patch to Front End Processors which issue requests for measurements and command signals. A typical LPS configuration consists of 22 HIMs to service GSE requirements. Interfacing with the HIMs and the vehicle systems are 14 FEPs. A fifteenth FEP is used to interface the CDBFR with other external systems. There are 15 consoles, organized on a subsystem basis, and two processors servicing the PDR/SPA system. The consoles provide the interface necessary for man-machine interaction during the Space Shuttle checkout.

### *Operations console subsystem*

This subsystem consists of two hardware modules, the main module and a support module. Figure 2 is a block diagram of a typical console subsystem. The main module includes three color cathode ray tubes, one three-channel display generator with associated cursor controls and keyboards, three Programmable Function Panels, and one safing panel. (A safing panel consists of a set of hard-wired command switches whereby volatile systems can be commanded to a safe condition.) The support module contains the console computer, its peripherals, and a printer/plotter. The main module is divided into three operator stations. Each station can be operated independently of the other two or any combination can support a common test function.

The checkout and monitor functions are performed in the console computers. The console subsystem provides the primary man-machine interface for all LPS operations.

Test procedures are written in the Ground Operations Aerospace Language (GOAL) [3, 4] by test engineers. The language allows GOAL test program compilation directly from procedures.

The GOAL application programs are initially loaded in each console computer and remain resident throughout a Shuttle processing cycle. During normal operation, each console operator has all required procedures on the console disk.

During the execution of any procedure, the operator may start or stop at any point, branch to any logical step within the procedure, or single-step through the procedure. The operator may choose to exception-monitor test data, where only data not within a defined set of limits are displayed. These limiting values may be changed dynamically by procedure or by entry from the keyboard.

Each console subsystem contains a subset of the CCMS data bank on the console disk. The data bank includes such things as measurement names, FEP assignments, interlock information, and software communication parameters.

### *Common Data Buffer*

This buffer is the communication center of the CCMS. The CDBFR has a capacity of 64 ports, and processors or custom interfaces can be attached to each of them. The capability exists for multiple CDBFR configurations to exist in any one Launch Control Center area. In addition, first-in, first-out buffer interfaces are attached directly to the address and data bus lines within the CDBFR to provide a direct data-read interface to a number of external devices in parallel. The standard interface from the CDBFR to the rest of the LPS subsystem is through one of 64 Buffer Access Cards. Each of the Buffer Access Cards is scanned in rotation by the scanner controller. Consequently, it provides the capability of transferring data from one CPU to another without encountering contention problems. It also provides the capability for one CPU to issue an interrupt to another.

### *Front End Processor*

The functions of the FEP are subdivided into three major areas: 1) PCM Telemetry, 2) Vehicle Launch Data Bus, and 3) Ground Support Equipment Hardware Interface Module.

The PCM (Pulse Code Modulation) Telemetry FEP processes measurement data received from the orbiter and communicates with other CCMS subsystems through the Common Data Buffer.

A PCM converter uses direct memory access techniques to transfer PCM data into FEP memory in a double buffer configuration. The converter generates an interrupt to the CPU when a complete minor frame (160 eight-bit words) has been transferred (every 10 milliseconds) and then begins utilizing the second portion of memory to transfer the next frame of data.

The FEP maintains current measurement status in the Common Data Buffer. The console computers acquire measurement data by reading this information from assigned CDBFR locations. The FEP contains a measurement descriptor table in which the entries contain dynamic parameters required to identify and process all assigned measurements.

The Launch Data Bus (LDB) FEP subsystem is required to process a variety of bi-directional data between General Purpose Computers (GPCs), on-board systems, and the Launch Processing System. The Launch Data Bus interfaces with two one-megabit-per-second Manchester-biphase-encoded (a standard encoding technique [5]) data buses from the vehicle, one active at a time.

The Launch Data Bus operates in a demand environment. Every bus transaction involving data transmission is caused by a request from a ground interface. When no transaction is required, the FEP answers a General Pur-

pose Computer (GPC) query (issued 40 milliseconds following the last transaction) with a no-message response.

The FEP is capable of transmitting bus commands independent of the power state of the orbiter's on-board computers. It is kept aware of the GPC power state since this parameter limits the on-board functional interfaces. Following is a brief description of the operational concepts depending on the state of GPC power.

In the GPC-on mode, the FEP responds to interrogate commands from the GPC. The GPC has direct control of the Launch Data Bus and communicates with the FEP by issuing an interrogate command 40 ms after the last bus transmission.

In the GPC-off mode, the FEP has direct control of the Launch Data Bus. Communication is available on either data bus independent of the 40-ms interrogate command cycle. This operational mode permits direct communication with on-board systems.

The Ground Support Equipment FEPs acquire and process measurement data from Hardware Interface Modules. Measurement processing includes significant data change detection and associated CDBFR status updates. Data are also checked for exception conditions and the FEP issues output commands to the Hardware Interface Modules.

The GSE FEPs are identical in architecture to provide ease of switching to a backup processor when needed. Microprogramming is used for many of the standard routines. The tables resident in the Front End Processor describe data to be monitored and may be changed on-line from the console Common Data Buffer interface. These tables are loaded in each FEP at system initialization.

The FEPs operate in conjunction with a discrete status matrix when issuing commands. A matrix is subsystem-oriented and maintained in each subsystem console. Required measurement status is maintained in the CDBFR and is continually monitored by any subsystem console requiring the measurement as an input to its matrix. Outputs from each matrix are also stored in designated locations in the CDBFR where they may be used as inputs to other matrices. When an FEP is requested to issue a command, the only check required is that the requestor be the proper subsystem console. The command will have been checked for status prerequisites and legality by the subsystem console.

The polling and processing function is performed as a time-sequenced event based on bus cycle times. Other background tasks, including CPU-to-CPU communication, measurement descriptor table updates, and self-tests, are executed in the FEPs asynchronously.

The GSE FEPs perform exception monitoring and the responsible console is notified when an exception condi-

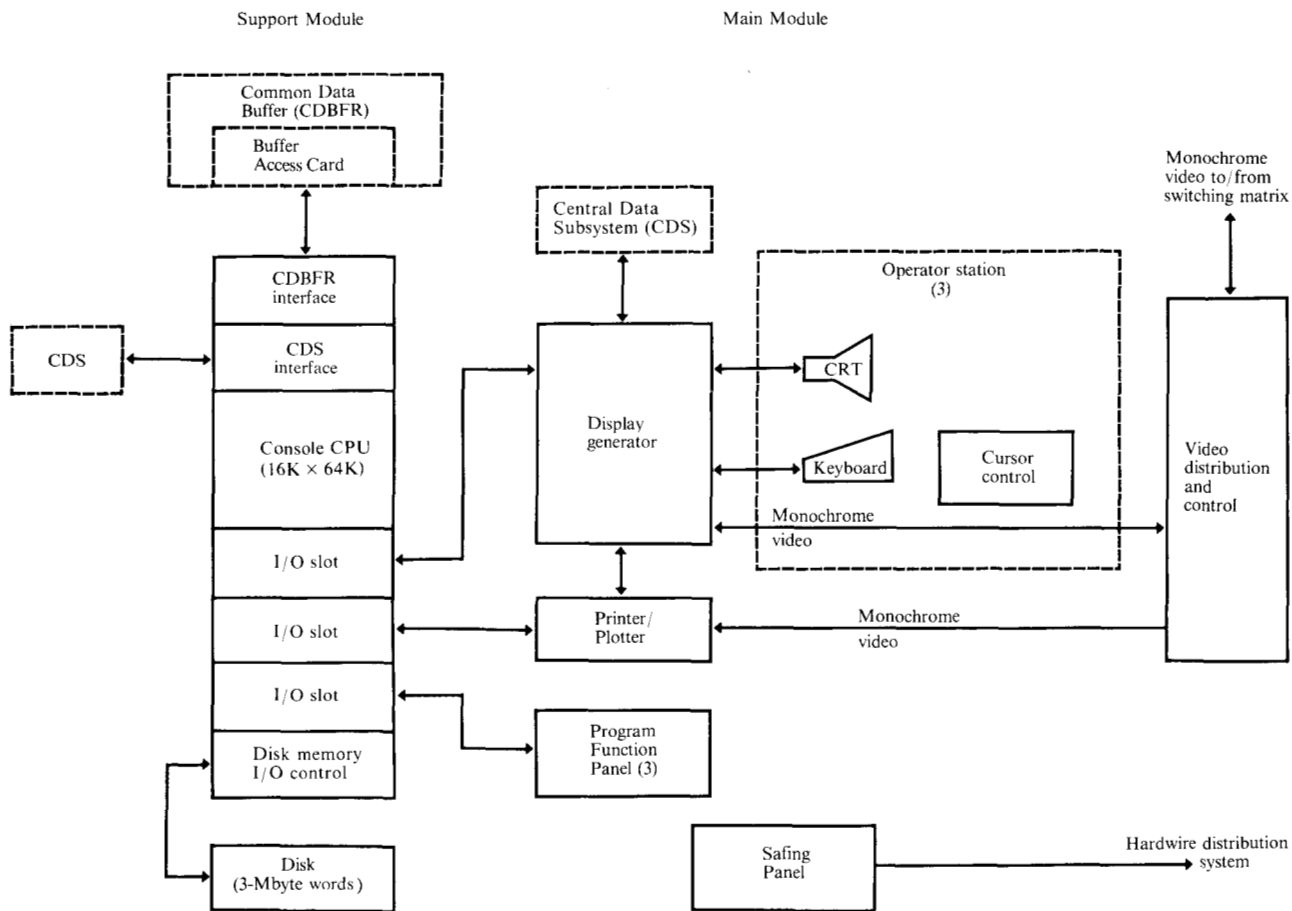


Figure 2 Operations console subsystem hardware.

tion is first detected. If the condition continues on subsequent samples, no additional notification is provided. Normal FEP processing of the measurement continues and its status is maintained in the Common Data Buffer. When the measurement returns to its normal condition, the responsible console is once again notified.

Real-time redundancy requirements are satisfied by redundant measurements assigned to different GSE data buses. A Front End Processor is not required to have any knowledge of these redundant measurements.

The GSE FEP throughput requirements limit data bus utilization. Current estimates of processing requirements indicate that 70 to 100 microseconds per measurement sample is the maximum required rate. One-hundred microseconds per measurement sample is equivalent to 10000 transmissions per second or an approximately 28-percent data bus utilization. In the event of an increase in measurement numbers or sample rate requirement, an FEP throughput limitation may occur. In that case, additional data buses and FEPs can be added. This precludes any subsystem overloading and takes advantage of the baseline FEP modular design philosophy.

#### PDR/SPA subsystem

This subsystem provides a means to record and play back measurement and command data. The Processed Data Recorder interfaces with the CDBFR through a first-in, first-out (FIFO) buffer. The PDR is primarily responsible for recording all data on magnetic tape. A secondary responsibility is maintaining 30 minutes of data available on disk for direct access.

Shared Peripheral Area functions include general peripheral processing for each of the computers. Specific application programs and system message data are routed to this area for output to line printers. Near real-time stripchart recording and plotting are made available via the shared peripheral processing function. All data are retained in a format that facilitates post-test analysis.

A time-oriented index is used to facilitate the data retrieval activity. The Shared Peripheral Area processor accesses the index to obtain the disk for the desired time interval. A utility program prints the index of tapes so that the tapes required for post-processing can be identified. Requests for data retrieval are time-oriented and originate at the console processor. The SPA accesses

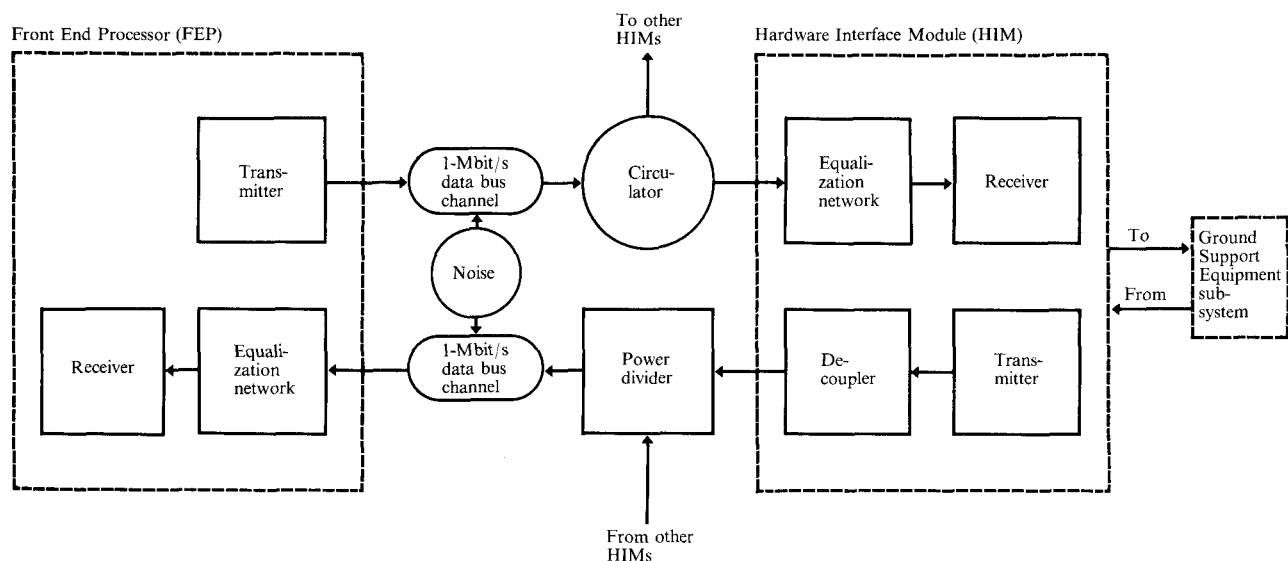


Figure 3 Ground Support Equipment transmit/receive system.

the data from the disk using a two-channel switch; it communicates with the PDR to obtain permission to access the disk, thereby minimizing interference. The data reduction function includes the capability of terminating the data recall in progress.

#### Hardware Interface Modules

These are the primary LPS interface terminals to the Ground Support Equipment. The HIM is an input/output device designed to provide monitoring and command capabilities that allow the CCMS to monitor and control the GSE.

The HIM uses a modular design concept, allowing different configurations to accommodate varying user requirements. This concept is implemented by using a family of standard stimulus/monitor cards. If the need arises, special stimulus/monitor cards can be provided to accommodate unique or nonstandard requirements.

On receipt of a data request, the selected HIM program scans the requested measurement and routes the value(s) back to the interrogating FEP. A command is acknowledged by the HIM by returning the issued command code in reply to a Front End Processor transmission. Any error is indicated to the FEP by an error response sync code, and the error status is maintained in the HIM status register.

#### Data transmission system

The data transmission system for the LPS consists of subsystems connecting the Checkout, Control, and Monitor Subsystem to GSE, vehicle, and other support systems.

The CCMS exchanges data with the Central Data Subsystem through two standard 9.6-kbit/s voice transmission systems and through the Common Data Buffer with a first-in, first-out buffer and a standard Buffer Access Card connection.

The transmission of data between GSE and the CCMS is accomplished by means of a 1-Mbit/s data bus; see Fig. 3. The data bus connects as many as eight Hardware Interface Modules to a Front End Processor. Bit-by-bit decoding is used and each bit must be decoded to a zero-one or a one-zero half-bit combination. Failure of either requirement or a parity error results in a transmission error detection. The probabilities of a) error, b) undetected error, c) expected error rate, and d) expected undetected error rate are shown in Table 1 for bit error rates of  $10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$ , and  $10^{-6}$ . An undetected error from Gaussian noise occurs only once in  $7.77 \times 10^{21}$  years.

Complete measurement and command redundancy are guaranteed by providing completely different paths for critical functions, including the measurement sensor and the command relay. Critical measurements are duplicated by redundant sensors, addressed through different FEP-HIM transmission paths with both measurements resident in the Common Data Buffer at different locations. Each of the measurements has a different address and measurement number. Each console dependent on a critical measurement is responsible for maintaining the redundant address list.

The transmission of data between the CCMS and the vehicle is accomplished through several standard-rate telemetry command links and a redundant, 1-Mbit/s

**Table 1** Probabilities of error in biphasic bit decoding with parity data-bus protocol.

	<i>Bit error rate</i>			
	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$
Probability of at least one error per one-way transmission	$1.5 \times 10^{-1}$	$1.5 \times 10^{-2}$	$1.5 \times 10^{-3}$	$1.5 \times 10^{-4}$
Probability of at least one undetected error per one-way transmission	$3.0 \times 10^{-34}$	$3.0 \times 10^{-46}$	$3.0 \times 10^{-58}$	$3.0 \times 10^{-70}$
Expected number of messages containing at least one error per 16-hour day for one complete transaction every				
70 $\mu$ s	$2.0 \times 10^8$	$2.4 \times 10^7$	$2.5 \times 10^6$	$2.5 \times 10^5$
100 $\mu$ s	$1.5 \times 10^8$	$1.7 \times 10^7$	$1.7 \times 10^6$	$1.7 \times 10^5$
Expected number of messages containing at least one undetected error per 16-hour day for one complete transaction every				
70 $\mu$ s	$5.0 \times 10^{-25}$	$5.0 \times 10^{-37}$	$5.0 \times 10^{-59}$	$5.0 \times 10^{-61}$
100 $\mu$ s	$3.5 \times 10^{-25}$	$3.5 \times 10^{-37}$	$3.5 \times 10^{-59}$	$3.5 \times 10^{-61}$
Time (years) between undetected errors for an average message interval of				
70 $\mu$ s	$7.7 \times 10^{21}$	$7.7 \times 10^{33}$	$7.7 \times 10^{45}$	$7.7 \times 10^{57}$
100 $\mu$ s	$1.1 \times 10^{22}$	$1.1 \times 10^{34}$	$1.1 \times 10^{46}$	$1.1 \times 10^{58}$

Launch Data Bus interfacing directly with the on-board computer system. The transmission scheme on the Launch Data Bus is similar to the GSE data bus in that Manchester biphasic [5] coding is used with bit-by-bit decoding. The data bus is normally under control of the on-board computers and a 40-ms interval is maintained between communications. The Launch Data Bus operates in a half-duplex mode over a single coaxial line. Consequently, the interface to the LDB Front End Processor is slightly different from that for the two-cable full-duplex GSE FEP system. In addition to bit-by-bit decoding and parity checking, block transmissions between the LDB FEP and on-board computers use a sum-check word for additional transmission security.

**Checkout Control, and Monitor Subsystem communication network**

The CCMS distributed computer system configuration requires a complex communication network to provide computer-to-computer communication. Problems encountered in the communication network design, such as contention, reliability, and stringent response time and throughput requirements, are common to the development of distributed computer system designs.

The Common Data Buffer (CDBFR) is the coordination point of all data communication in the CCMS configuration. The CDBFR contains 32K (32768), expandable to 64K, 16-bit words of memory. The first thousand locations are used for interrupt-control words, required to

facilitate computer-to-computer communication. The second thousand locations make up the common read/write area, accessible by all computers. The remainder of the Common Data Buffer storage is used for CPU private write areas. All CPUs have the ability to access all CDBFR locations.

The first-in, first-out high speed buffer at the Processed Data Recorder and/or Central Data Subsystem ports to the CDBFR interprets a log signal and reads the CDBFR address and data buses at the FIFO interface. Therefore, logging can be selectively requested for any word written into the CDBFR and accomplished in parallel with the write operation without any increase in CDBFR operations. Each CPU on the Common Data Buffer is directly connected to a port through a Buffer Access Card. Each port can be activated or placed in the standby mode under software control and assigned a write-protected area of storage from controls on the CDBFR control panel. The activate/standby control lines, available at each port, prevent a computer from accessing the CDBFR. This feature can be used to prevent a malfunctioning computer from either storing data in erroneous locations or storing erroneous data.

Each port is scanned sequentially by the scanner controller, located in the CDBFR. If no access from a port is required, the scanner controller interrogates the next port after a delay of less than 50 nanoseconds. High speed memory modules are used to provide the required speed. If an access is required by a port, either a read or

Table 2 Minimal computer-to-computer communication formats.

---

<i>Transaction format</i>	
Function code	
Source-computer identification	
Destination-computer identification	
Transaction serial number	
Transaction data	
<i>Response format</i>	
Sender's transaction serial number	
Data to be placed in sender's buffer	

---

a write cycle takes place in approximately 250 ns. Meanwhile, a look-ahead capability is used to find the next port requiring service. A fully loaded 64-port configuration can be serviced in less than 16  $\mu$ s.

Any computer can send a transaction to any other computer by placing data in the transmitting computer's write-protected area and writing the starting address of the data to be transmitted into the interrupt-control area of the receiving computer. The receiving computer processes the interrupt, reads the required data, and sends an acknowledgment back to the transmitting computer. The acknowledgement indicates that the transmitted data were received. Any response required from processing the transmitted data requires a new request to be generated from the receiving computer.

The primary computer-to-computer functions are to acknowledge receipt of transactions from other computers, route transactions to appropriate processing tasks, send transactions to other computers, receive response data from other computers and move it from the CDBFR to the sending task's buffer, and assure that sending tasks do not wait indefinitely for acknowledgment of transactions sent to other computers.

Computer-to-computer communication processes the transmission requests by using application programs and interrupts from other computers. Four sub-components perform the processing: Send Supervisor Call (SVC) Routine, Receive Interrupt Handler, Computer-to-Computer Task, and Watchdog Task. The Watchdog Task is executed on a cyclic basis. By examining transaction serial numbers, it detects bottlenecks caused by the failure of a destination computer to acknowledge a transaction within the cyclic period of the task.

A computer-to-computer transaction is initiated by the sender, who constructs at least the minimal transaction format (Table 2) in his private write-protected area. The required transaction data may be any number of words. On completion of writing the transaction data into the assigned contiguous CDBFR locations, the sender issues an interrupt to the receiving computer. The inter-

rupt controller, contained in the CDBFR, handles the interrupt, establishes the proper CDBFR location, and enables the proper address lines, allowing the issuing computer to write the "pointer" word in the proper location.

The pointer word contains the starting address of the transaction data. The interrupt controller "points" the receiving computer to the proper Common Data Buffer location in order to read the pointer word.

Transactions are kept intact in the sending computer's private write-protected area until the receiving computer acknowledges receipt of the transaction. An acknowledgment is another computer-to-computer communication but it points to the original sender's transaction. Consequently, the sender can establish that this is an acknowledgment of a pending transaction rather than a new transaction.

In some cases, a response may be required after an action is completed. In this case, the original receiving computer builds another computer-to-computer communication in its private write-protected area and issues an interrupt with the proper pointer word to the original sending computer. The transaction data contain the data necessary to classify the impending transaction as a response and to identify the previous transaction. The response data have a minimal format, also identified in Table 2.

### Summary

The Launch Processing System Checkout, Control, and Monitor Subsystem enables the Shuttle checkout to take place within a two-week period with a minimum of operation personnel interaction. Checkout personnel interface with their particular subsystem through the use of one of 15 operations consoles, each of which can command and monitor specific subsystem tests and support fully operational integrated system tests. A special console is used for system integration while another monitors the operation and reliability of the CCMS hardware.

The use of a high speed Central Data Buffer enables the interconnection of up to 64 minicomputers to perform the checkout, control, and monitor functions required by the Space Shuttle. The distributed nature of this computing system imposes severe requirements for computer-to-computer communication. The communication network design philosophy described in this paper may have direct application to other distributed computer systems.

### Acknowledgment

The system designs described in this paper were developed with the National Aeronautics and Space Administration, John F. Kennedy Space Center, on contract NAS10-8580.



## References

1. "Launch Processing System Station Set 84 Requirements Document," *K-SM-10-123*, Shuttle Project Office, NASA John F. Kennedy Space Center, Florida, April 22, 1974.
2. "LPS Concept Description Document," *KSC-DD-LPS-007*, Directorate of Design Engineering, John F. Kennedy Space Center, Florida, January 11, 1974.
3. "Ground Operations Aerospace Language," Contract NAS 10-6900 Final Report, Vol. 1: Study Overview, IBM Federal Systems Division, Cape Canaveral, Florida, July 31, 1973.
4. "Ground Operations Aerospace Language (GOAL)," *TR 1228*, NASA John F. Kennedy Space Center, Florida, September 1, 1974.
5. "Military Standard—Aircraft Internal Time Division Multiplex Data Bus," *MIL-STD-1533 (USAF)*, August 30, 1973.

*Received February 16, 1975; revised July 22, 1975*

*Mr. Byrne is located at the NASA John F. Kennedy Space Center, Florida 32815; Mr. Doolittle is with the IBM Federal Systems Division, 150 Sparkman Drive NW, Huntsville, Alabama 35805; and Mr. Hockenberger is with the IBM Federal Systems Division, Cape Kennedy Facility, 7900 North Astronaut Boulevard, Cape Canaveral, Florida 32920.*