MCDONNELL
DOUGLAS

CONTRACT NAS8-25140
MSFC-DPD-235/DR NO. SE-02

INFORMATION MANAGEMENT SYSTEM STUDY RESULTS

VOLUME I
IMS Study Results

NOVEMBER 1971

MDC G2584

APPROVED BY:

Vern D Kirkland

VERN D. KIRKLAND
PROGRAM INTEGRATION/OPERATIONS
DIRECTOR
SPACE STATION PROGRAM

(NASA-CR-123547) INFORMATION MANAGEMENT SYSTEM STUDY RESULTS. VOLUME 1: IMS STUDY RESULTS (McDonnell-Douglas Astronautics Co.) Nov. 1971 515 p CSCL 05B

N72-19972

G3/34

Unclas
15112

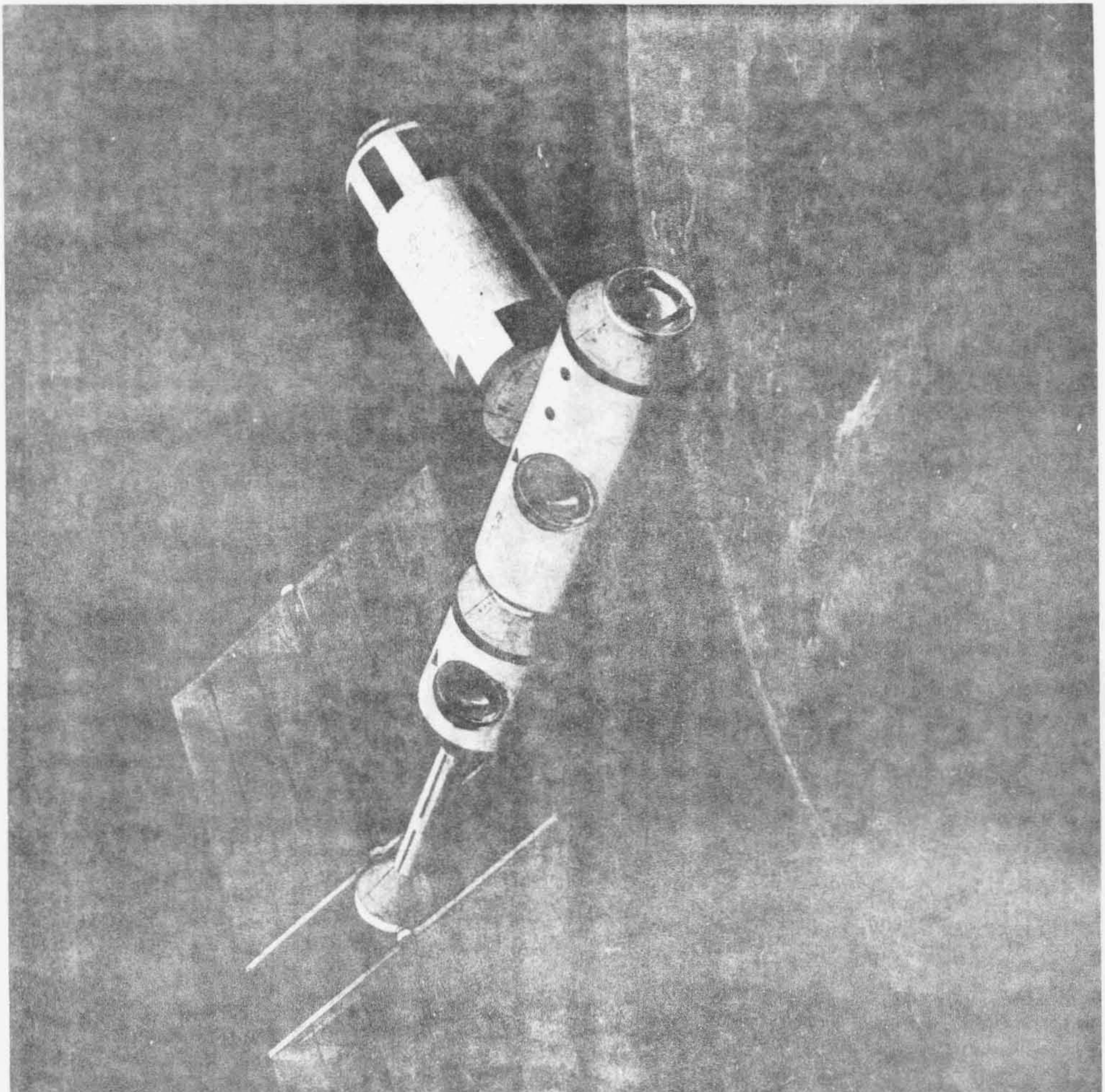
(NASA CR OR TMX OR AD NUMBER) (CATEGORY)

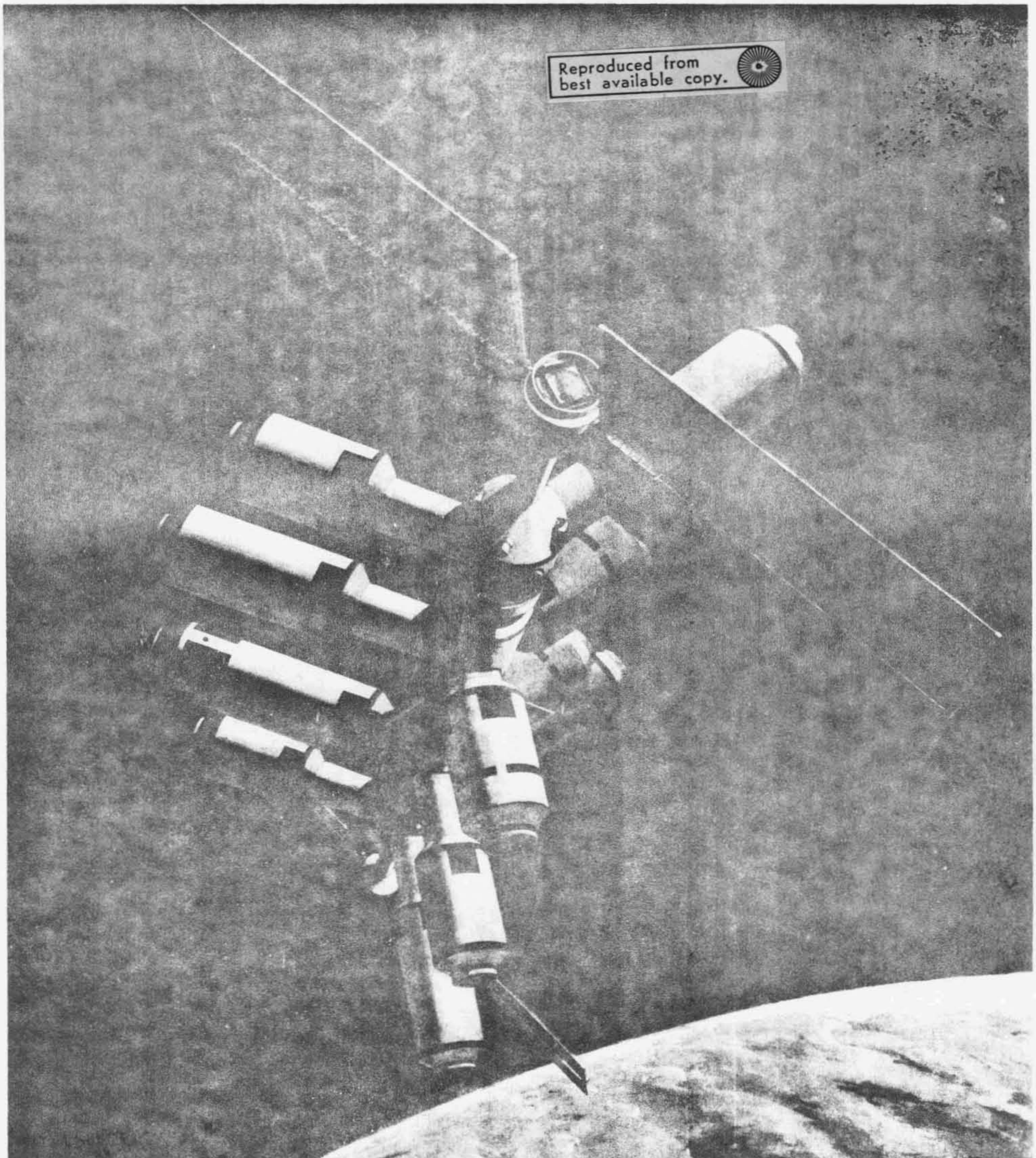
FF No

[redacted]

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY
5301 Bolsa Avenue, Huntington Beach, CA 92647

516





PREFACE

The work described in this document was performed under the Space Station Phase B Extension Period Study (Contract NAS8-25140). The purpose of the extension period has been to develop the Phase B definition of the Modular Space Station. The modular approach selected during the option period (characterized by low initial cost and incremental manning) was evaluated, requirements were defined, and program definition and design were accomplished to the depth necessary for departure from Phase B.

The initial 2-1/2 month effort of the extension period was used for analyses of the requirements associated with Modular Space Station Program options. During this time, a baseline, incrementally manned program and attendant experiment program options were derived. In addition, the features of the program that significantly affect initial development and early operating costs were identified, and their impacts on the program were assessed. This assessment, together with a recommended program, was submitted for NASA review and approval on 15 April 1971.

The second phase of the study (15 April to 3 December 1971) consists of the program definition and preliminary design of the approved Modular Space Station configuration.

A subject reference matrix is included on page v to indicate the relationship of the study tasks to the documentation.

This report is submitted as Data Requirement SE-02.

DATA REQUIREMENTS (DR's)
MSFC-DPD-235/DR NOs.
(contract NAS8-25140)

Category	Designation	DR Number	Title
Configuration Management	CM	CM-01	Space Station Program (Modular) Specification
		CM-02	Space Station Project (Modular) Specification
		CM-03	Modular Space Station Project Part 1 CEI Specification
		CM-04	Interface and Support Requirements Document
Program Management	MA	MA-01	Space Stations Phase B Extension Study Plan
		MA-02	Performance Review Documentation
		MA-03	Letter Progress and Status Report
		MA-04	Executive Summary Report
		MA-05	Phase C/D Program Development Plan
		MA-06	Program Option Summary Report
Manning and Financial	MF	MF-01	Space Station Program (modular) Cost Estimates Document
		MF-02	Financial Management Report
Mission Operations	MP	MP-01	Space Station Program (Modular) Mission Analysis Document
		MP-02	Space Station Program (Modular) Crew Operations Document
		MP-03	Integrated Mission Management Operations Document
System Engineering and Technical Description	SE	SE-01	Modular Space Station Concept
		SE-02	Information Management System Study Results Documentation
		SE-03	Technical Summary
		SE-04	Modular Space Station Detailed Preliminary Design
		SE-06	Crew/Cargo Module Definition Document
		SE-07	Modular Space Station Mass Properties Document
		SE-08	User's Handbook
		SE-10	Supporting Research and Technology Document
		SE-11	Alternate Bay Sizes

SUBJECT REFERENCE MATRIX

LEGEND:
CM Configuration Management
MA Program Management
MF Manning and Financial
MP Mission Operations
SE System Engineering and Technical Description

	CM				MA		MF	MP			SE								
	CM-01 Space Station Program (Modular) Specification	CM-02 Space Station Project (Modular) Specification	CM-03 Modular Space Station Project Part 1 CEI Spec	CM-04 Interface and Support Requirement Document	MA-05 Phase C/D Program Development Plan	MA-06 Program Option Summary Report	MF-01 Space Station Program (Modular) Cost Estimates Document	MP-01 Space Station Program (Modular) Mission Analysis Document	MP-02 Space Station Program (Modular) Crew Operations Document	MP-03 Integrated Mission Management Operations Document	SE-01 Modular Space Station Concept	SE-02 Information Management System Study Results	SE-03 Technical Summary	SE-04 Modular SS Detailed Preliminary Design	SE-06 Crew/Cargo Module Definition Document	SE-07 Modular Space Station Mass Properties Document	SE-08 User's Handbook	SE-10 Supporting Research and Technology	SE-11 Alternate Bay Sizes
2.0 Contractor Tasks																			
2.1 Develop Study Plan and Review Past Effort (MA-01)																			
2.2 Space Station Program (Modular) Mission Analysis								●											
2.3 Modular Space Station Configuration and Subsystems Definition														●					
2.4 Technical and Cost Tradeoff Studies							●												
2.4.4 Modular Space Station Option Summary											●								
2.5 Modular Space Station Detailed Preliminary Design														●					
Mass Properties																●			
2.6 Crew Operational Analysis									●										
2.7 Crew Cargo Module															●				
Mass Properties																●			
2.8 Integrated Mission Management Operations										●									
2.9 Hardware Commonality Assessment													●						
2.10 Program Support					●														
2.11 Requirements Definition																			
Space Station Program (Modular)	●																		
Space Station Project (Modular)		●																	
Modular Space Station Project—Part 1 CEI			●																
Interface and Support Requirements				●															
2.12 Plans					●														
2.13 Costs and Schedules							●												
2.14 Special Emphasis Task Information Management (IMS)												●							
Modular Space Station Mass Properties																●			
User's Handbook																	●		
Supporting Research and Technology																		●	
Technical Summary													●						
MOD 29																			●
MOD 40						●													

CONTENTS

FIGURES

TABLES

Section 1
INTRODUCTION

1.1 BACKGROUND

With the advent of the Space Shuttle in the late 1970's, a long-term manned scientific laboratory in Earth orbit will become feasible. Using the shuttle for orbital buildup, logistics delivery, and return of a scientific data, this laboratory will provide many advantages to the scientific community and will make available to the United States a platform for application to the solution of national problems such as ecology research, weather observation and prediction, and research in medicine and the life sciences. It will be ideally situated for Earth and space observation, and its location above the atmosphere will be of great benefit to the field of astronomy.

This orbiting laboratory can take many forms and can be configured to house a crew of up to 12 men. The initial study of the 33-foot-diameter Space Station, launched by the Saturn INT-21 and supporting a complement of 12, has been completed to a Phase B level and documented in the DRL-160 series. Recently completed studies are centered around a Space Station comprised of smaller, shuttle-launched modules. These modules could ultimately be configured to provide for a crew of the same size as on the 33-foot-diameter Space Station—but buildup would be gradual, beginning with a small initial crew and progressing toward greater capability by adding modules and crewmen on a flexible schedule.

The Modular Space Station Phase A—level study results are documented in the DRL-231 series. Recent Modular Space Station Phase B study results are documented in the DPD-235 series, of which this is a volume.

The Space Station will provide laboratory areas which, like similar facilities on Earth, will be designed for flexible, efficient changeover as research and experimental programs proceed. Provisions will be included for such functions as data processing and evaluation, astronomy support, and test and calibration of optics. Zero gravity, which is desirable for the conduct of experiments, will be the normal mode of operation. In addition to experiments carried out within the station, the laboratories will support operation of experiments in separate modules that are either docked to the Space Station or free-flying.

Following launch and activation, Space Station operations will be largely autonomous, and an extensive ground support complex will be unnecessary. Ground activities will ordinarily be limited to long-range planning, control of logistics, and support of the experiment program.

The Initial Space Station (ISS) will be delivered to orbit by three Space Shuttle launches and will be assembled in space. A crew in the Shuttle orbiter will accompany the modules to assemble them and check interfacing functions.

ISS resupply and crew rotation will be carried out via round-trip Shuttle flights using logistics modules (Log M's) for transport and on-orbit storage of cargo. Of the four Log M's required, one will remain on orbit at all times.

Experiment modules will be delivered to the Space Station by the Shuttle as required by the experiment program. On return flights, the Shuttle will transport data from the experiment program, returning crewmen, and wastes.

The ISS configuration rendering is shown in the frontispiece. The power/subsystems module will be launched first, followed at 30-day intervals by the crew/operations module and the general purpose laboratory (GPL) module. This configuration will provide for a crew of six. Subsequently, two additional modules (duplicate crew/operations and power/subsystems modules) will be mated to the ISS to form the Growth Space Station (GSS) shown in the frontispiece), which will house a crew of 12 and provide a capability equivalent to the 33-foot INT-21-launched Space Station. GSS logistics support will use a crew cargo module capable of transporting a crew of six.

During ISS operations, five research applications modules (RAM's) will be assembled to the Space Station. Three of these will be returned prior to completion of the GSS. In the GSS configuration, 12 additional RAM's will augment the two remaining from the ISS phase. Three of the RAM's delivered to the GSS will be free-flying modules.

During the baseline 10-year program, the Space Station will be serviced by Shuttle-supported logistics module or crew cargo module flights.

1.2 SCOPE OF THIS VOLUME

The Information Management System (IMS) Special Emphasis Task was performed as an adjunct to the Modular Space Station Study, with the objective of providing extended depth of analysis and design in selected key areas of the information management system. Specific objectives included the following:

A. Perform in-depth studies of IMS requirements and design approaches.

B. Design and fabricate breadboard hardware for demonstration and verification of design concepts.

C. Provide a technological base that will identify potential design problems and influence long-range planning.

D. Develop hardware and techniques to permit long-duration, low-cost manned space operations.

E. Support SR&T areas where techniques or equipment are considered inadequate.

F. Permit an overall understanding of the IMS as an integrated component of the Space Station.

Major contributions to the task in the computer and data bus areas were made by IBM under subcontract to MDAC.

Specific tasks are summarized below.

1.2.1 Computer

Design studies were performed to define, to the block diagram level, simplex and multiprocessor computer systems for the Space Station. The task also included definition of the input/output system, an optimized instruction set, and an executive program for the computer. The MSFC SUMC processor was used as the basis for these designs.

1.2.2 Data Bus

Design studies were performed to define the data bus system. These consisted of analysis of data transmission requirements, and investigation of alternate design concepts in key areas, such as bus control techniques, modulation and encoding methods, and signal distribution. Fabrication and delivery of a working breadboard model consisting of two data bus terminals and two remote data acquisition units are included in the task.

1.2.3 Displays and Controls

This task provides for design, fabrication, and delivery of a working breadboard model of the Space Station display and control console. The breadboard is designed to interface with the data bus and contains representative hardware such as CRT displays, audio cues, light emitting diode displays, lights, meters, switches, and alphanumeric/function keyboards.

1.2.4 Communications

Working breadboard models of a K_u-band exciter and rf power amplifier are provided.

1.2.5 Packaging and Installation

Studies were performed to develop an equipment packaging and installation design for the Space Station. This design includes features providing for standardization of equipment design and enhancement of on-orbit maintainability. Fabrication and delivery of a nonfunctional installation mockup is included in the task. In addition, the deliverable Remote Data Acquisition Units (part of the data-bus system) are designed in mockup-compatible form to demonstrate the packaging concepts relative to functional equipment.

1.2.6 Checkout Language

A definition was developed for a higher-order test-engineering-oriented computer language for versatile communication and comprehension of test programs.

1.2.7 System Integration

Overall test plans for the IMS breadboard were developed, including hardware and software requirements.

1.2.8 Reliability and Maintainability

Reliability and maintainability features of the IMS design were documented.

Section 2
COMPUTER

The IMS Special Emphasis Task produced design studies and analysis in several key areas associated with the Space Station computer system. These included block-diagram-level definition of the data-processing system and the computer input/output system, and definition of an optimized instruction set for the computer. A preliminary design specification for an Executive Program was also developed. All hardware and software designs were based upon the requirements of the Modular Space Station, and were predicated upon utilization of the MSFC Space Ultrareliable Modular Computer (SUMC) processor. The results of these studies are contained in the following sections.

2.1 COMPUTER SYSTEM DESIGN

2.1.1 Data-Processing Requirements

The data processing requirements developed from the Modular Space Station portion of the study were used as a guideline in defining the computer system design requirements. These requirements are discussed in Sections 4.10.2 and 4.10.3.1.1 of the Modular Space Station Report (SE-04). Data processing requirements for the subsystem and experiment support facilities are summarized in Table 2-1. The quantities of computer facilities assemblies required to perform these requirements are summarized in Table 2-2. These quantities were based on the baseline SUMC computer, input/output controller, and data bus performance characteristics, as well as projections in main, auxiliary, and bulk memory technology advances. These requirements were used in defining the multiprocessor system configuration (Section 2.1.3), the multiprocessor input/output controller configuration (Section 2.2.3), the multiprocessor instruction set (Section 2.3) and the Executive Program requirements (Appendix A).

Table 2-1

DATA-PROCESSING REQUIREMENTS SUMMARY

	Processing Rate (KOPS)		Main Memory (K words)		Auxiliary Memory (K words)	
Facility	ISS(1)	GSS(2)	ISS	GSS	ISS	GSS
Subsystem	727	855	105	128	885	1044
Experiment	486	779	80	128	486	789

Notes: (1) ISS: Initial Space Station.
(2) GSS: Growth Space Station.

The Modular Space Station as currently envisioned does not require the simplex computer system configuration. However, the simplex computer system configuration was used as a baseline for multiprocessor efforts and may be considered a subset of the multiprocessor system. The simplex configuration could be used as an independent system for other applications, or could be used for special experiment processing applications if the need arises later in the Space Station Program.

2.1.2 Simplex System Configuration

The purpose of this section is to:

A. Establish a baseline of computer and peripheral hardware and associated software for a simplex computer system configuration centered around the MSFC SUMC central processor unit (CPU).

B. Establish the basis for growth from a simplex configuration to multiprocessor configurations involving several processors.

C. Define major interfaces between CPU, input/output controller (IOC), main memory, auxiliary memories, bulk memories, and data bus elements.

D. Define changes required to the SUMC to accommodate IOC operations.

Table 2-2

COMPUTING ASSEMBLIES SUMMARY

Assembly	ISS Operating	ISS Spares (powered down)	GSS Operating	GSS Spares (powered down)
Subsystem Facility				
CPU's	2	1	2 to 3	1
IOC's	1	1	1	1
Main memory (16 K-word modules)	7	2	8	2
Auxiliary memory (1 M-word modules)	1	1	2	1
Bulk memory	2	1	2	1
Experiment Facility				
CPU's	2	1	2 to 3	1
IOC's	1	1	1	1
Main memory (16 K-word modules)	5	2	8	2
Auxiliary memory (1 M-word modules)	1	1	1	1
Bulk	2	1	2	1

Note: K = 1024

M = 1,048,576

2.1.2.1 Simplex Computing System Overview

The Simplex Computing system includes:

A. MSFC SUMC CPU.

B. Msin memory modules.

C. Input/output controller.

D. Auxiliary memory units.

E. Bulk memory units.

The major simplex computing system elements and their respective interfaces are indicated in Figure 2-1. The central processor unit is the MSFC SUMC currently under development. The input/output controller design requirements are defined in the I/O design requirements, Section 2.2.1. The main memory modules, auxiliary memory, and bulk memory devices are summarized in Section 2.1.2.2. The major interface between the computer and experiments and subsystems supported by the computing system is through the I/O controller and the data bus elements, including the data bus coax cable, data bus terminals, and remote data acquisition units (RDAU's).

2.1.2.2 Simplex System Elements

Block diagrams and summary definitions of simplex computing system elements are contained in the following paragraphs.

Central Processor Unit (CPU)

The block diagram for the CPU (Figure 2-2) is similar to that of the MSFC SUMC described in MSFC Report SP-232-0384, "MSFC Advanced Aerospace Computer," dated 6 July 1970. The floating point unit (FPU), arithmetic logic unit (ALU), scratch-pad memory (SPM), control unit (CU), and multiplexer register unit (MRU) are summarized in Section 2.0 of the referenced document. Modifications will be required for restriction of privileged instructions to executive program control, storage protection, input/output control, and other single or multiple-level interrupts. These modifications are described in Section 2.1.3 of this document.

Input/Output Controller

Figure 2-3 is the IOC baseline for the simplex computer system. The IOC is divided into two blocks: (1) computer interface unit (CIU) and (2) subsystem interface unit/computer (SIU/C). Functions of each block in the CIU are summarized as follows:

A. Program Control contains the input/output multiplexers, channel status indicators, timing and controls and interface logic to the

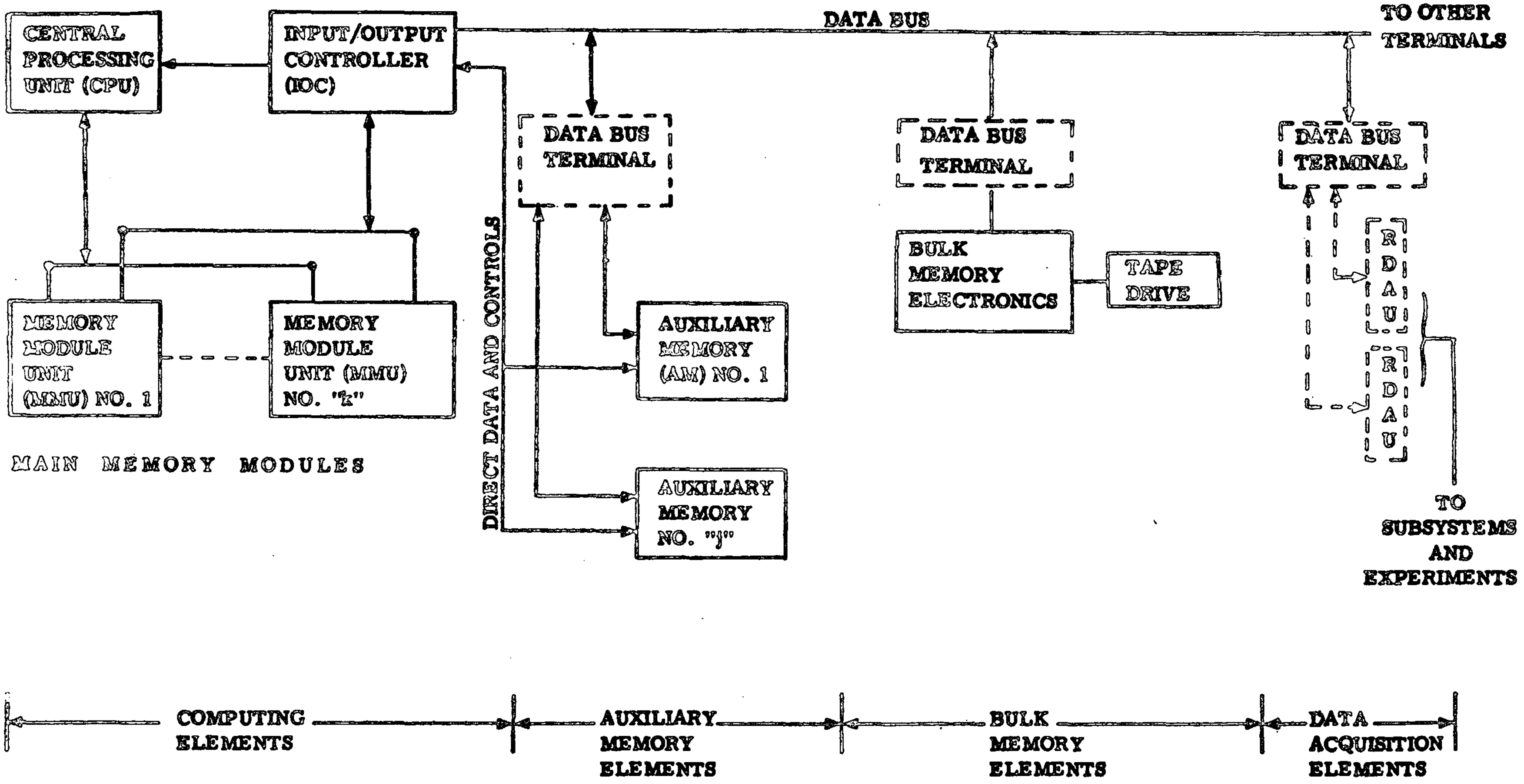


Figure 2-1. Simplex System Block Diagram

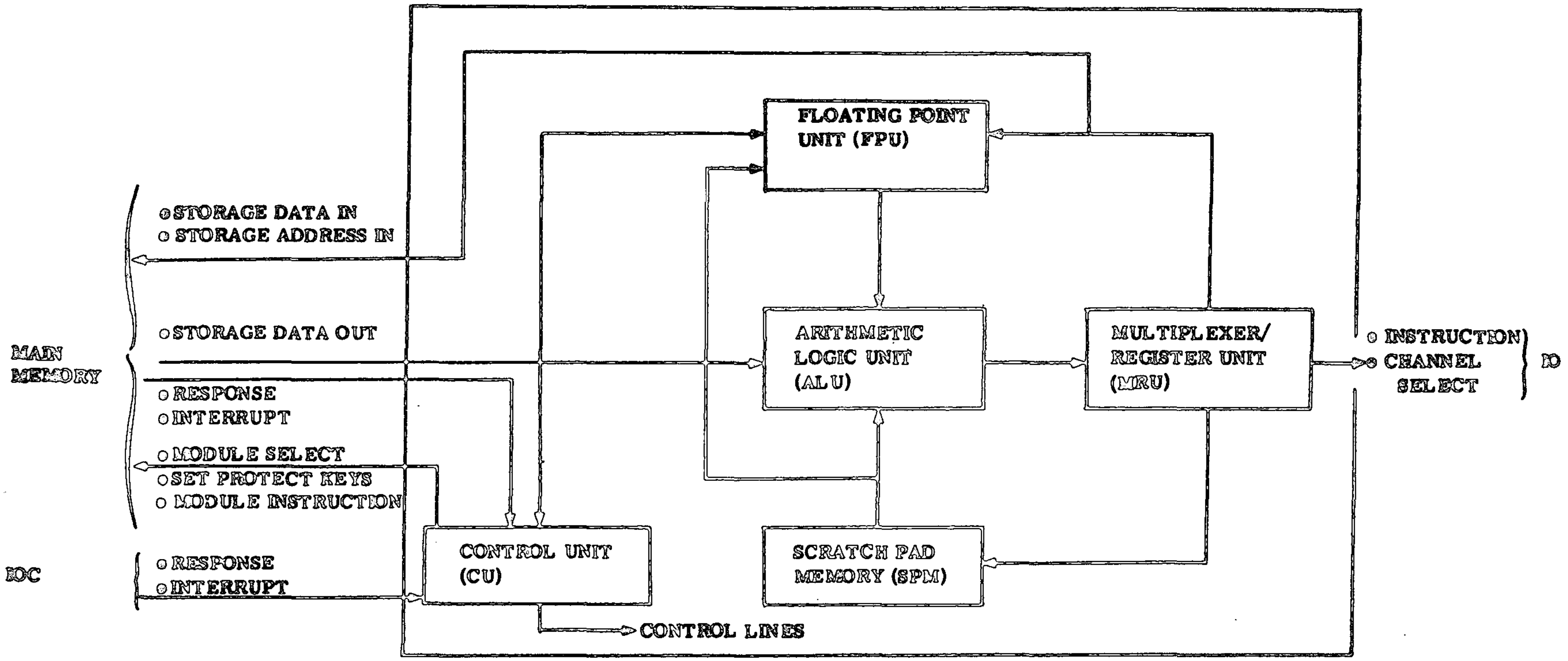


Figure 2-2. Simplex CPU Block Diagram

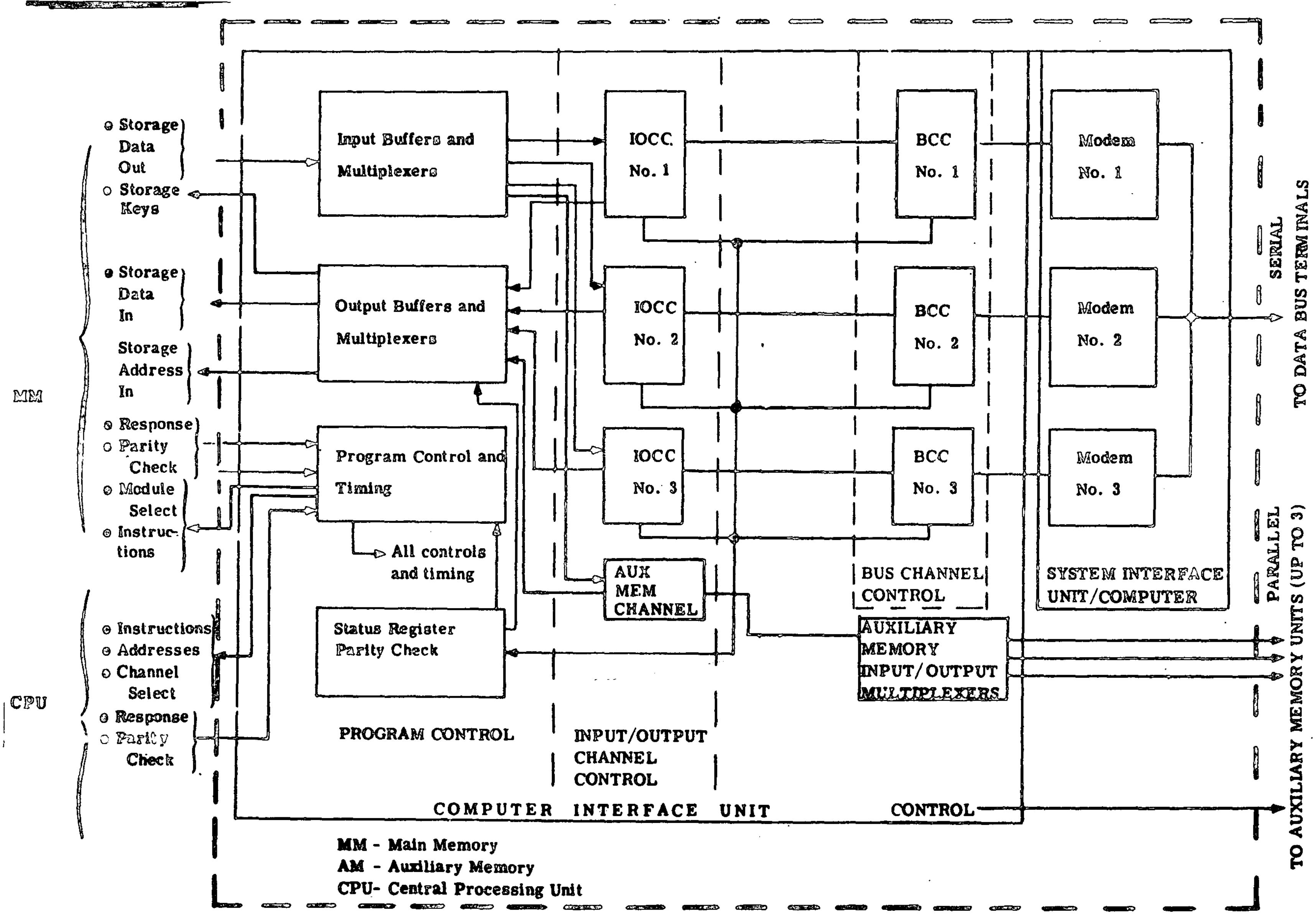


Figure 2-3. Simplex I/O Controller

computer, and the logic to recognize and initiate the action requested.

B. I/O Channel Control accomplishes the bidirectional data transfers between memory and bus channel control units and between the main memory and auxiliary memory units. The IOC controls the addressing and sequencing of information flow between the main memory modules and the bus channel control or auxiliary memory units. The basic logic is duplicated for each channel and is made up of address and count registers, register increment logic, and associated control logic.

C. Bus Channel Control includes data buffers, serialize and deserialize logic, and interface logic to control data flow between I/O channel and a moden unit.

The SIU/C consists of three modems. Each modem operates on a carrier frequency separated from the other modems. Each modem uses the same frequency for both transmitting and receiving.

Main Memory

The main memory consists of low-power, high-speed modules compatible with the CPU and IOC. The main memory contains the executive program and active data and applications programs. Auxiliary programs and data may be transferred into the main memory from the auxiliary or bulk memory units.

Each memory module will have a unique four-bit address to identify the module which will be decoded in the CPU or IOC to select the particular module to be addressed. Random-access addressing will be used at the module level and within modules.

Each module will contain provisions to protect storage areas that cannot be read out or written into without authorization. Each module will have the capability to recognize read or write requests which have access to protected storage areas.

Each module will have an input/output port to interface with the CPU and a second input/output port to interface with the IOC. The memory modules will use hybrid bipolar metal-oxide-semiconductor technology because of low power and weight, potentially high reliability, and volume and cost considerations. The main memory module concept with input/output ports and storage protection features is illustrated in Figure 2-4.

Auxiliary Memory

The auxiliary memory consists of a medium-capacity, medium-speed unit for temporary storage of data and instructions which may be utilized by the computing elements on demand. The auxiliary memory may also be used to buffer the transfer of data between elements on the data bus.

The auxiliary memory unit will be organized into blocks which may be accessed randomly. Access within a block will be sequential. Data transfers in and out of each block will be via parallel words.

The auxiliary memory will interface directly with the IOC in order to exchange data with the CPU and main memory. The auxiliary memory will interface with the data bus through data bus terminal for the temporary storage and buffering of data exchanged with the data bus.

The auxiliary memory will use nonmetallic magnetic domain (bubble) technology because of potentially high reliability, low power, weight, volume and cost considerations.

The auxiliary memory concept is illustrated in Figure 2-5.

Bulk Memory

The bulk memory unit is a ultrahigh-density tape recorder that is organized to sequentially read in and read out parallel words from and to the data bus.

The bulk memory unit is a high-capacity, high data-rate unit used to store large quantities of digital data for subsequent transmission to the ground, either physically or by communications links. The bulk memory may also

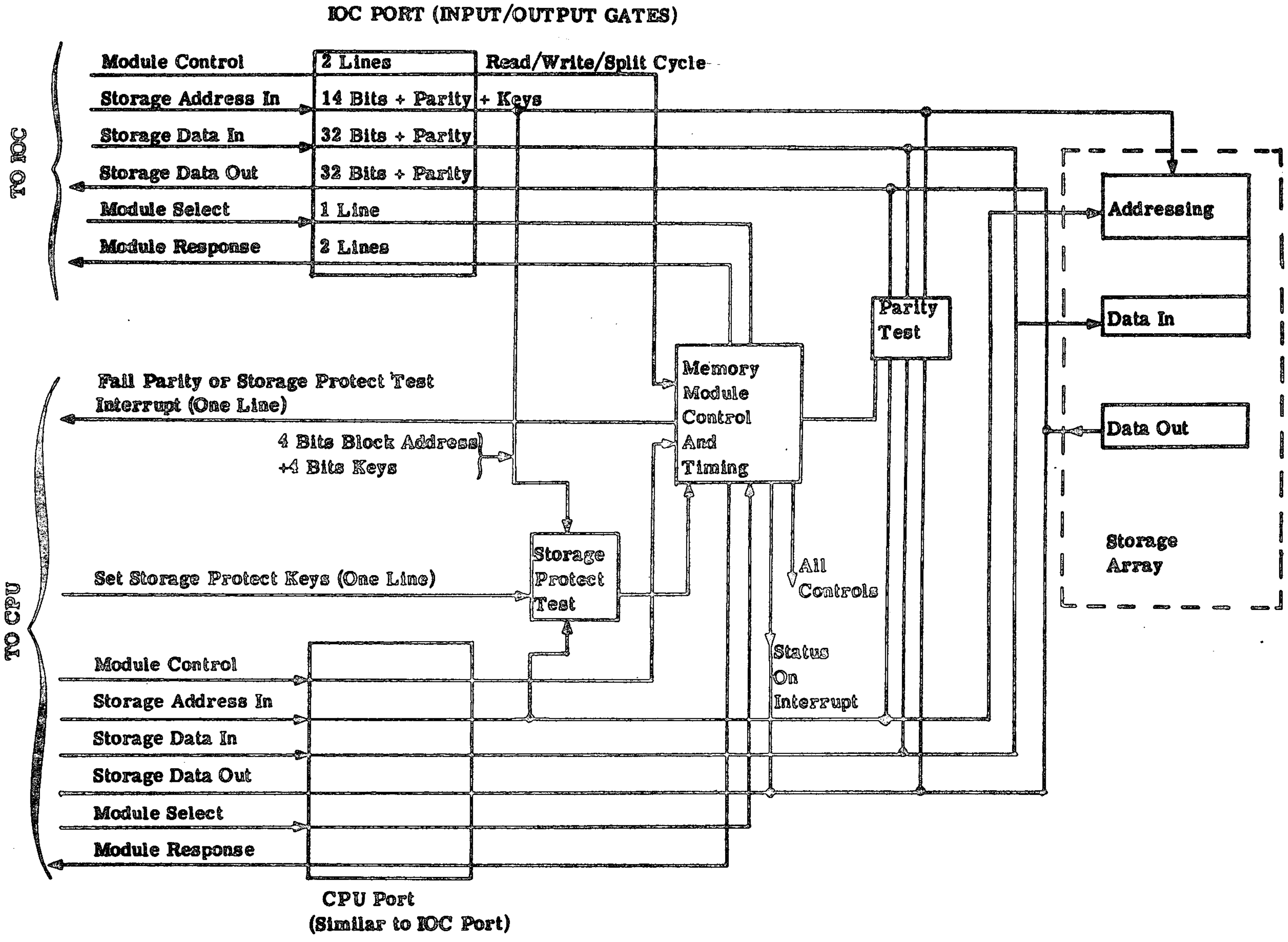


Figure 2-4. Main Memory Module Concept

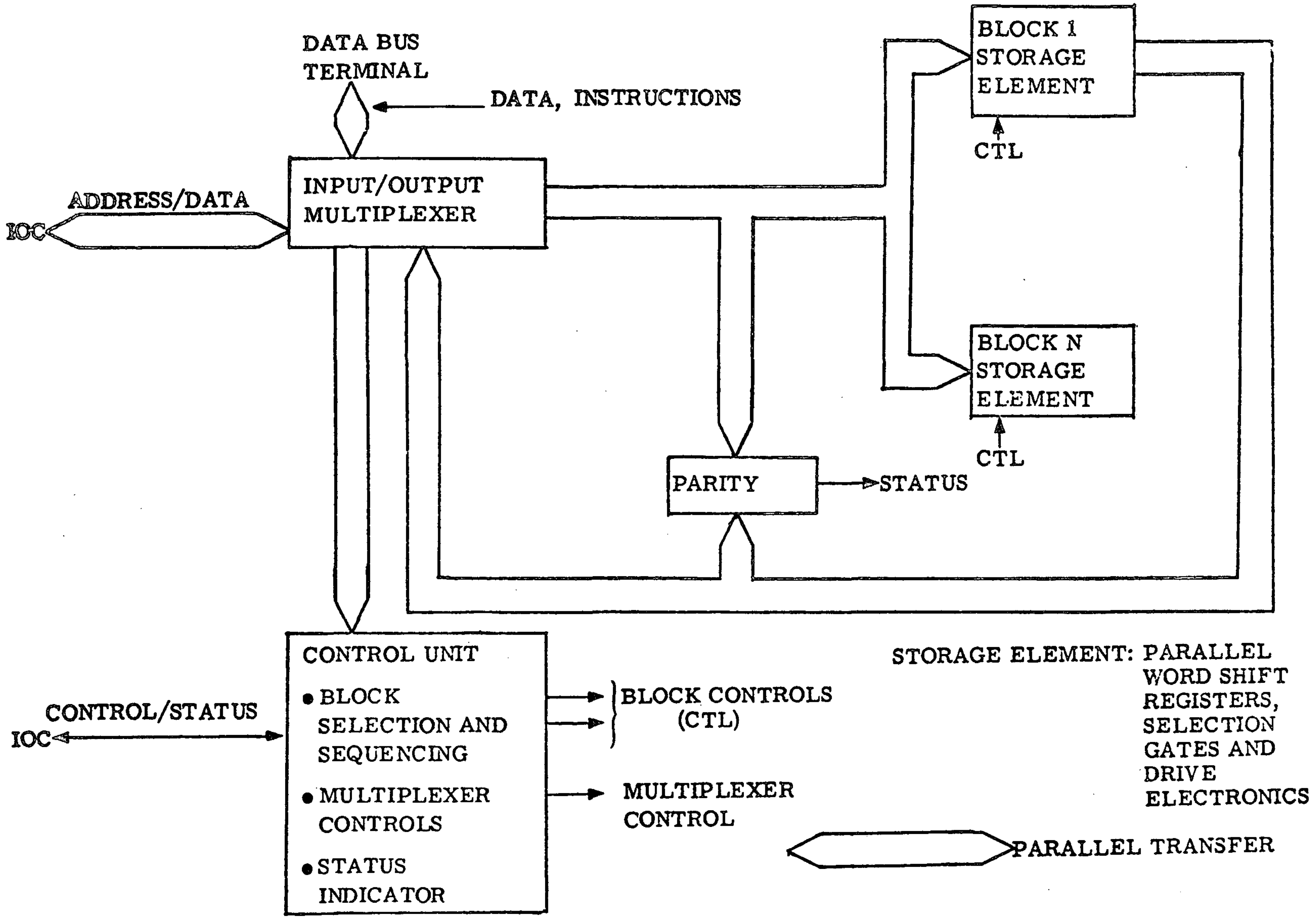


Figure 2-5. Auxiliary Memory Concept

provide infrequently used data and instructions to the computing elements from data stored in Space Station files as selected by the crew.

The data bus terminal provides transmitting and receiving interfaces between the bulk-memory electronics and the data bus. The data bus terminal provides buffering to match the data bus rates to the tape drive record and reproduce capabilities. Commands from the data bus are decoded and used to control the record and reproduce electronics and tape drive controller functions.

The tape drive unit contains the record and reproduce heads, drive motors, and associated mechanisms.

2. 1. 2. 3 Data and Control Flow Between Major Elements

CPU to I/O Controller

The CPU issues commands under CPU executive program control to the IOC to initiate a particular control sequence or list of sequences.

IOC to CPU

The IOC provides an interrupt signal to the CPU when operations have been terminated on any data bus channel or when the parity test has failed. The IOC provides its operational status to the CPU upon request.

CPU to Main Memory

Under microprogram control the CPU issues memory operation code, memory address, and information to be stored (if a store operation). The CPU sets up storage-protect keys in each memory module, as established by the executive program. The CPU selects the particular module in main memory to be addressed.

Main Memory to CPU

The main memory will provide status, instructions, and data to the CPU in accordance with CPU requests. Each main memory module will send an interrupt signal to the CPU for storage protect and parity test failures.

IOC to Main Memory

The IOC will supply the main memory with the memory operations to be performed and the addresses of a sequence of I/O operations to be executed. The IOC will write into main memory data received from the data bus as directed.

Main Memory to IOC

The main memory will provide response and parity check signals to the IOC followed by a sequence of commands to be executed by the IOC. The main memory will provide data to the IOC for transmission to the data bus or auxiliary memory.

IOC to Data Bus

The IOC will issue commands and data to the data bus. Commands will be used to control data bus device operations. Data will be provided to the data bus for transfer to storage or communications devices, or to control subsystem and experiment operations.

Data Bus to IOC

The data bus data stream will contain status information from terminals, RDAU's and other devices, and data to the IOC in response to IOC commands.

IOC to Auxiliary Memory

The IOC will provide commands, instructions, and data to auxiliary memory devices. The IOC will directly interface with the auxiliary memory for faster access and reduction of data bus traffic. Commands for power on/off, read, write, and other switching purposes will be issued.

Auxiliary Memory to IOC

The auxiliary memory will provide status information, instructions, and data to the IOC. The auxiliary memory will interface directly with the IOC.

I/O Controller to Bulk Memory

The I/O controller will provide commands, instructions, and data to the bulk memory devices through the data bus. Commands will be for power on/off, read, write and other switching purposes.

Bulk Memory to I/O Controller

The bulk memory will provide status information, instructions, and data to the I/O controller through the data bus.

Data Bus to Auxiliary Memory

The data bus will provide instructions and data to the auxiliary memory unit via a data bus terminal. Typical information to be transferred consists of infrequently used instructions and data from bulk memory, or experiment data which requires buffering before processing, bulk storage, or RF communication to the ground.

Auxiliary Memory to Data Bus

The auxiliary memory unit will provide instructions and data to the data bus via a data bus terminal. Typical information to be transferred includes buffered data for bulk storage or transmittal to ground station via rf.

2. 1. 3 Multiprocessor System Configuration

2. 1. 3. 1 Introduction

This section of the report presents the results of a task to configure a Space Station multiprocessor utilizing the SUMC processor. A companion task to configure the I/O for the multiprocessor is presented subsequently. The multiprocessor description is divided into two primary sections: Section 2. 1. 3. 2 is a summary of the configuration and its features, and Section 2. 1. 3. 3 is a more detailed description of the design. A glossary of abbreviations is given in Table 2-3.

Table 2-3
ABBREVIATIONS

A	accumulator
ALU	arithmetic logic unit
AM	auxiliary memory
ATR	address translate register
ATU	address translate unit
B	base
CCR	configuration control register
CCU	configuration control unit
CPU	central processing unit
CSR	communications state register
CSU	communications state unit
D	displacement
DBT	data bus terminal
FPM	floating point multiplexer
FPU	floating point unit
IAROM	instruction address read only memory
IC	iteration counter
ILR	initial load reset
INSTR	instruction
I/O	input/output
IOCC	input/output channel controller
IOC	input output controller
IR	instruction register
MAM	memory address multiplexer

Table 2-3

ABBREVIATIONS (Continued)

MAR	memory address register
MM	main memory
MMU	main memory unit
MPX	multiplexer
MQM	multiply/quotient multiplexer
MQR	multiply/quotient register
MR	memory register
MROM	microprogram read only memory
MRU	multiplexer/register unit
PC	program counter
PRM	product/remainder multiplexer
PRR	product/remainder register
PSA	preferential storage area
PSR	preferential storage register
ROM	read only memory
RSMU	reconfiguration state monitor unit
SCU	sequencer control unit
SPM	scratch-pad memory
SUMC	space ultrareliable modular computer (SUMC abbreviations have been used as defined in Baseline Documents and included herein.)
X	index
Z	absolute address (D+X+B)

2.1.3.2 Summary

General Features

Figure 2-6 shows the multiprocessor block diagram. As described herein a maximum configuration would consist of:

4 processors (CPU's)
15 memory modules (MM's)
3 input/output controllers (IOC's)

Fewer elements of each type can be used although it would not be a multiprocessor unless two or more CPU's were used, and several MM's would be expected. Similarly, more units of each type could be accommodated by the design concept but the addressing would have to be expanded.

The multiprocessor has the following general features:

A. Any CPU or IOC can directly address any MM module.

B. The executive program can reconfigure the system to accommodate changing needs or hardware malfunction, if it is so designed.

C. Because of the shared memory, several CPU's can work from a common set of separate program units without "rolling" programs in and out of main memory modules.

D. Selected groups of hardware elements can be configured into an independent subsystem for maintenance/checkout, software debug, bringing up new subsystems, etc.

E. Base register addressing facilitates relocating programs in memory, or double indexing.

F. Hardware memory module address translation divorces logical module addresses from hardware addresses allowing contiguous program addressing independent of the particular memory modules actually in use.

G. Use of standardized, general interfaces facilitates the addition or interchange of CPU's, memories, or I/O hardware to or in a particular configuration.

H. Storage keys minimize the impact of software or hardware addressing errors by protecting software memory blocks from unauthorized access.

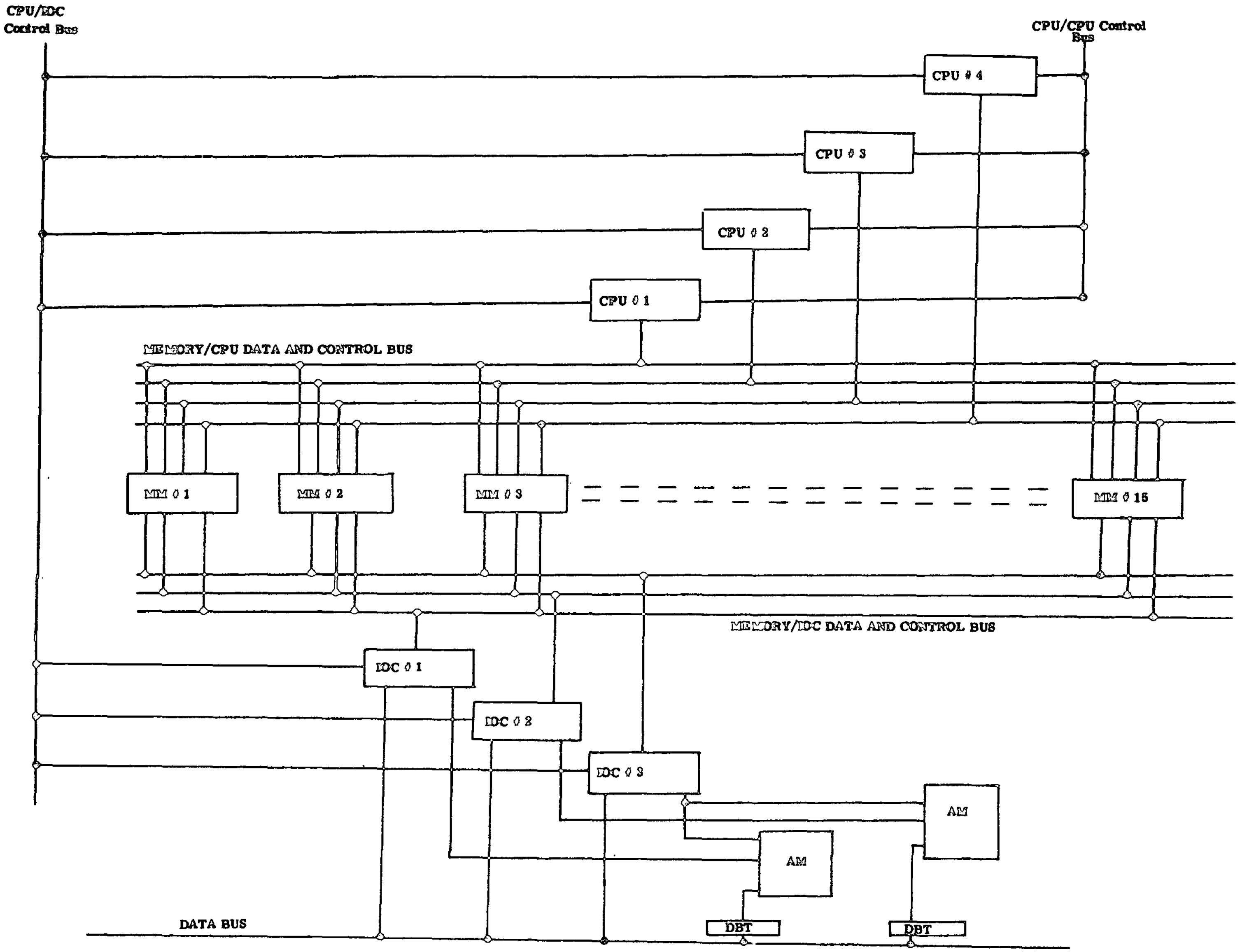


Figure 2-6. Multiprocessor Block Diagram

I. Memory access timeouts prevent CPU or IOC hangups from memory failures.

J. Multiprocessing oriented instructions will be implemented in the microprogram control storage.

K. If any CPU goes into the reconfiguration state, all other CPU or IOC operation is inhibited by hardware, so that reconfiguration commands and data can use normal data paths.

Operation

Multiprocessor Operation—The primary operating mode of the system is as a multiprocessor. In this mode the tasks will be entered into the system through some man/machine interface such as a keyboard. The executive control program will then determine the configuration required to meet the processing.

As each processor within the multiprocessor configuration completes a task, it will utilize the task assignment portion of the executive program to determine the next task assignment. A TEST AND SET instruction is used to insure that only one CPU is performing a single-specific task at a time. As a task is assigned, the CPU will obtain the necessary storage keys, base register values, address translation data, and interrupt/privileged instruction mask words to allow successful execution of the task.

Four basic states are provided for the control of the operating modes of the multiprocessor elements, as shown in the following table.

State	Mode	Options
0	Off-line	1. Test* — Can be recalled via state 3 CPU. 2. Isolated** — Manually isolated via interface isolate switch.
1	Nonoperating	1. Powered up but halted (enables power-down command).

State	Mode	Options
2	Operating	Normal operating state.
3	reconfiguration	CPU Only—Enables execution of Load CCU External instruction.

*In this mode, the element is not available for normal multiprocessor use.
**In this mode, the executive control program cannot alter the CCU in any way.

Typical experiment scheduling will require a varying processor capability. As the need for new configurations arises, one processor under executive control will establish the new configuration. Since reconfiguration (state 3) halts all operation outside the reconfiguring CPU, the normal reconfiguration sequence would cause the other CPU's and the IOC's to establish a condition where they can halt without bad effects on system operation; i. e., I/O operations, etc., will be shut down. When a satisfactory quiescent state has been achieved, the reconfiguring computer will be instructed to enter state 3 by the executive and reconfigure the system. Since reconfiguration is achieved by a single computer, a reconfiguration state monitor unit (RSMU) is provided to safeguard against failures during reconfiguration.

Abnormal Operation—Normal reconfiguration has been described as an orderly process leading to a relatively quiescent state before any CPU enters state 3. However, abnormal operation resulting from hardware failures or software inadequacies can make an orderly shutdown impossible, and that is why the dominance of state 3 has been provided. It allows removal of an uncooperative hardware element from the system. In that case, some rollback to a previous checkpoint will be necessary to pick up the operations of the failed unit.

General Characteristics

CPU's — The CPUs in the multiprocessor will be the SUMC CPU's, with minor changes to the control storage area and with a configuration control unit (CCU) added. The CCU performs the following tasks:

A. Control the CPU state—operating, powered down, off-line, or reconfiguration.

B. Enable/inhibit communication with other CPU's, memories, and IOC's.

C. Translate logical (program) addresses to hardware addresses.

D. Provide hardware starting address for preferential storage area in main memory.

Main Memory — The main memory of the multiprocessor can be any storage technology so long as the speed is compatible with the CPU performance requirements; i. e. , if an add time of 2. 0 μs is required, a memory cycle must be 1. 0 μs or less. The memory/CPU interface is asynchronous, so that CPU's and memories with varying performance characteristics can be mixed without hardware impact. To facilitate this, the MMU's will contain address and data registers. The MMU's will have the following characteristics:

A. 16 K-word capacity.

B. 32 data bits plus parity.

C. A separate storage section for storage keys.

D. A multiport interface (one for each CPU and IOC).

E. A configuration control unit having state control and communication control similar to that of the CPU.

Input/Output — All I/O operations are controlled by the IOC. Once initiated by the CPU, the IOC works with the MMU's and data busses without requiring CPU activity to control I/O operations. CPU/IOC communication can be by CPU interrogation, IOC-generated interrupt, or via memory locations. The IOC's are described in a separate section of this report.

Fault Detection

Fault detection is a complex topic, and it is not within the scope of this investigation to give a detailed treatment of the subject; however, there are some comments which should be made about multiprocessor-unique aspects of fault detection.

A. In the normal multiprocessor mode, standard self-test software and hardware can be augmented by redundant elements to aid in detection and isolation of faults. Using good units to aid in testing bad units improves coverage and confidence in self-test.

B. This multiprocessor advantage is lost during the simplex reconfiguration operation. Therefore, a reconfiguration state monitor unit (RSMU) is provided as special test hardware to detect major malfunctions during reconfiguration.

The RSMU has the following features:

A. Fail-safe by duplex implementation.

B. Testable by the CPU's.

C. Detects multiple state 3's.

D. Detects excessive time in state 3.

E. Activates alarms and generates an interrupt to the CPU's, but does not initiate reconfiguration or shut down the system.

A more detailed description of the RSMU is given in the system description part of this report.

2.1.3.3 System Description

SUMC Review

Since the multiprocessor is to be based on the Space Ultrareliable Modular computer (SUMC) under development at MSFC, those features of SUMC of interest will be presented. This SUMC description is based on Sperry Rand Report No. SP-232-0384, "MSFC Advanced Aerospace Computer," July 1970, and discussions with R. C. Asquith of MSFC.

Of particular interest are the SUMC control unit, registers and data paths, and scratch-pad memory. A block diagram of SUMC is shown in Figure 2-7.

A. Control—The SUMC is controlled via a microprogram read only memory (MROM) which contains 1024 80-bit words.

B. Scratch-Pad Memory (SPM)—A 64-word by 32-bit scratch-pad memory is used for general arithmetic registers, base registers, index registers, the instruction counter, and temporary storage.

C. Data Paths—The flow of data through the SUMC can be seen by referring to the block diagram. These data paths are not altered for application of the SUMC to a multiprocessor configuration. Registers in the multiplexer/register unit (MRU) hold data for storage, SPM, the floating point unit, and the control unit.

It should be noted that the outputs of the MROM and SPM are stable as long as the input address is held. This sometimes relieves the need for registers in other parts of the logic.

The SUMC block diagram shows that the instruction translation is a two-step operation: (1) the eight-bit operation code addresses a 256-word by 11-bit read only memory. (2) Ten of the bits from the instruction address read only memory (IAROM) are used as the initial address of the MROM for execution of the instruction. The eleventh bit from the IAROM tells whether a memory cycle is required for obtaining a second operand from main memory.

Two possible instruction formats are shown in Figure 2-8. The sequence of events in the control unit (CU) associated with the fetching and execution of a typical instruction is shown in Figure 2-9. The actual instruction execution is shown as block 5 under control of the MROM. As can be seen by the loop from block 7 back to block 4, two or more control words from MROM can be chained to perform complex instructions, and the instruction set can be changed by altering (customizing) the contents of MROM and IAROM.

Figure 2-10 shows a more detailed sequence for the instruction fetch cycle. The read cycle shown during ROM cycle 4 is for the second operand and is

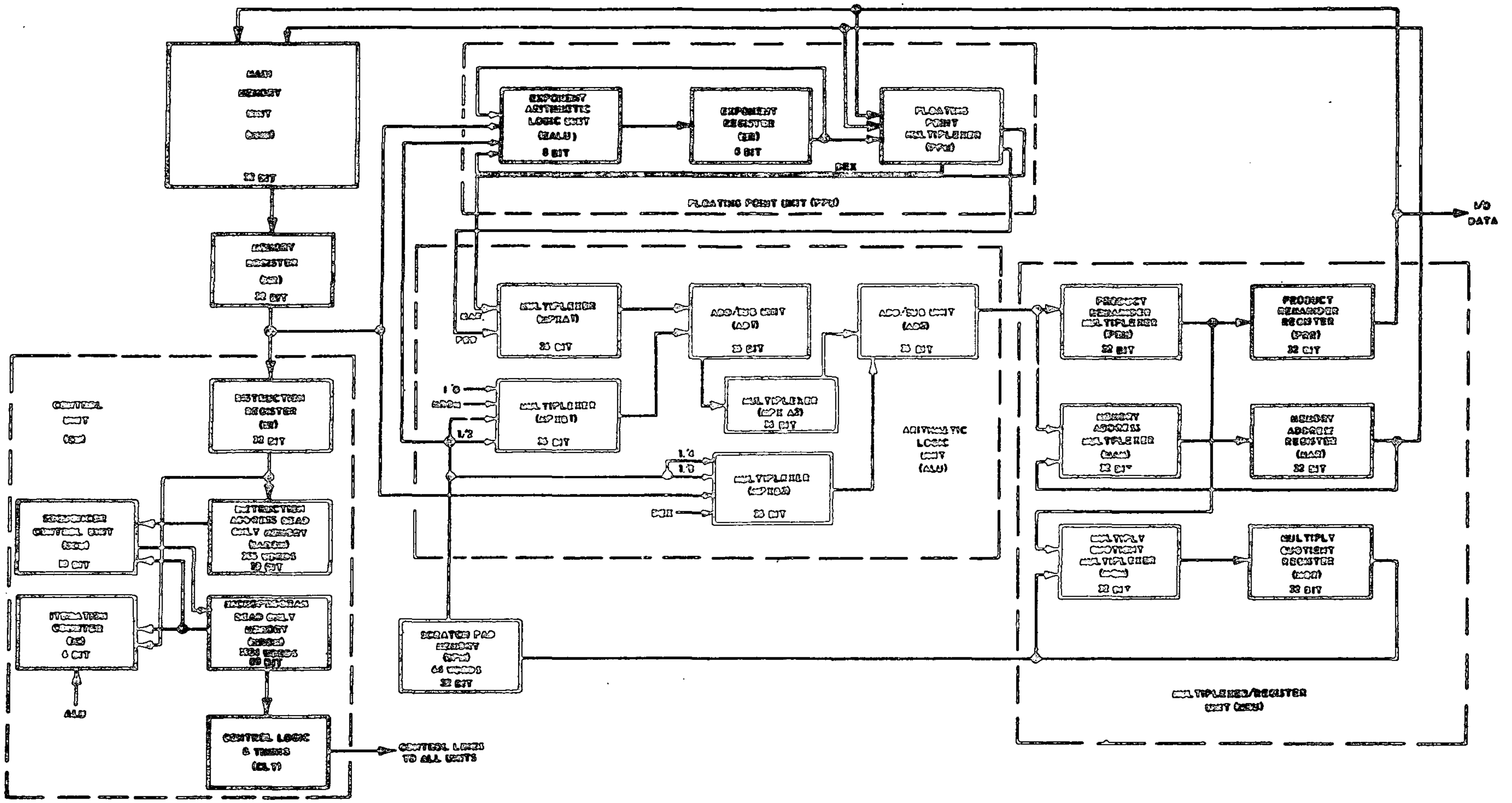

MAIN MEMORY UNIT (MMU) 32 BIT
MEMORY REGISTER (MR) 32 BIT
INSTRUCTION REGISTER (IR) 32 BIT
CONTROL UNIT (CU)
SEQUENCE CONTROL UNIT (SCU) 10 BIT
INSTRUCTION ADDRESS READ ONLY MEMORY (IAROM) 256 WORDS 10 BIT
ITERATION COUNTER (IC) 6 BIT
ALU
MICROPROGRAM READ ONLY MEMORY (MROM) 1024 WORDS 80 BIT
CONTROL LOGIC & TIMING (CLT)
CONTROL LINES TO ALL UNITS
SCRATCH PAD MEMORY (SPM) 64 WORDS 32 BIT
EXPONENT ARITHMETIC LOGIC UNIT (EALU) 8 BIT
EXPONENT REGISTER (ER) 8 BIT
FLOATING POINT MULTIPLEXER (FPM)
DEX
FLOATING POINT UNIT (FPU)
MAR
PRR
MULTIPLEXER (MPXA1) 33 BIT
ADD/SUB UNIT (AD1) 33 BIT
ADD/SUB UNIT (AD2) 33 BIT
I/O
MROM
1/2
MULTIPLEXER (MPXB1) 33 BIT
MULTIPLEXER (MPX A2) 33 BIT
1/4
1/8
MQR
MULTIPLEXER (MPXB2) 33 BIT
ARITHMETIC LOGIC UNIT (ALU)
PRODUCT REMAINDER MULTIPLEXER (PRM) 32 BIT
PRODUCT REMAINDER REGISTER (PRR) 32 BIT
MEMORY ADDRESS MULTIPLEXER (MAM) 32 BIT
MEMORY ADDRESS REGISTER (MAR) 32 BIT
MULTIPLY QUOTIENT MULTIPLEXER (MQM) 32 BIT
MULTIPLY QUOTIENT REGISTER (MQR) 32 BIT
MULTIPLEXER/REGISTER UNIT (MRU)
I/O DATA

Figure 2-7. Detailed Block Diagram, SUMC

]

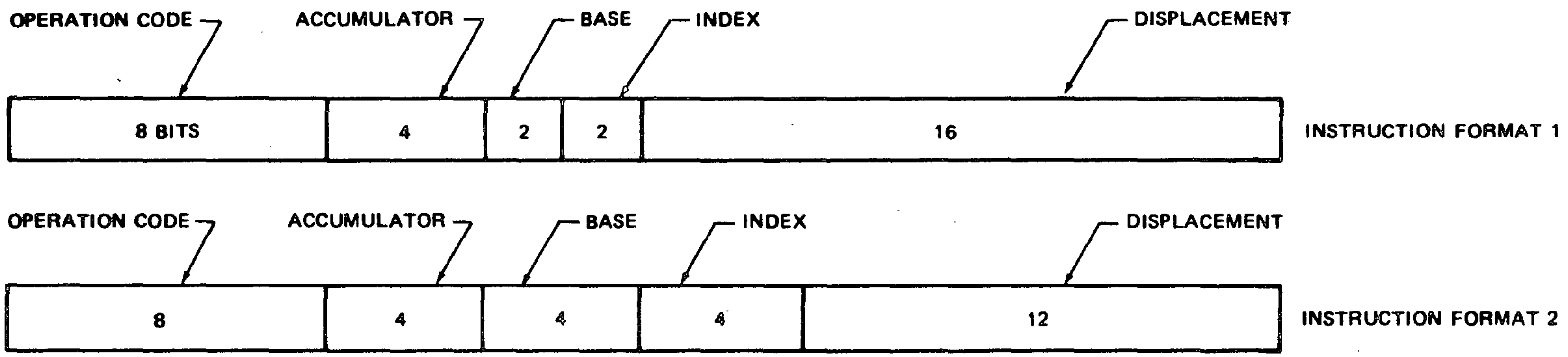


Figure 2-8. Instruction Formats

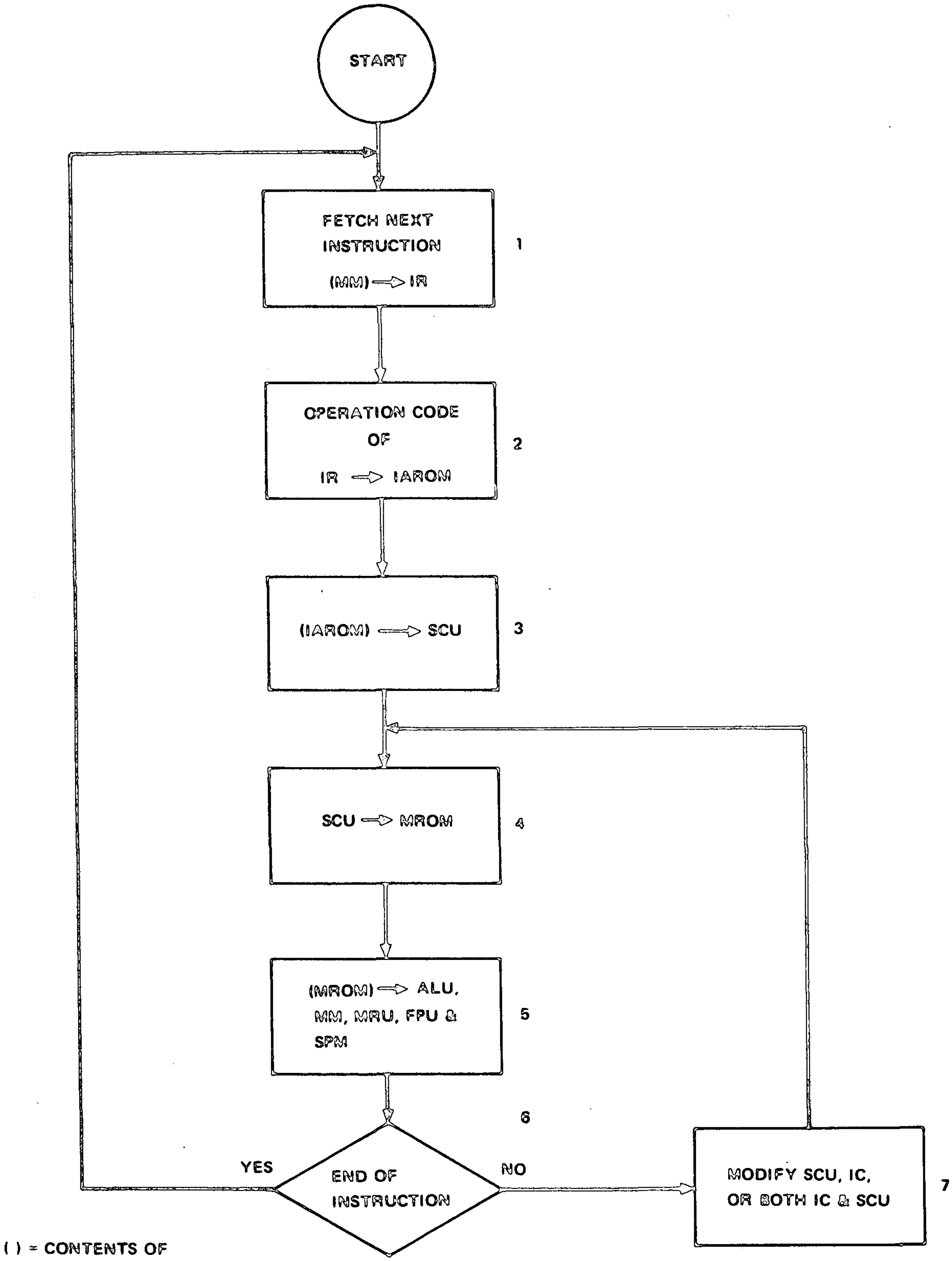
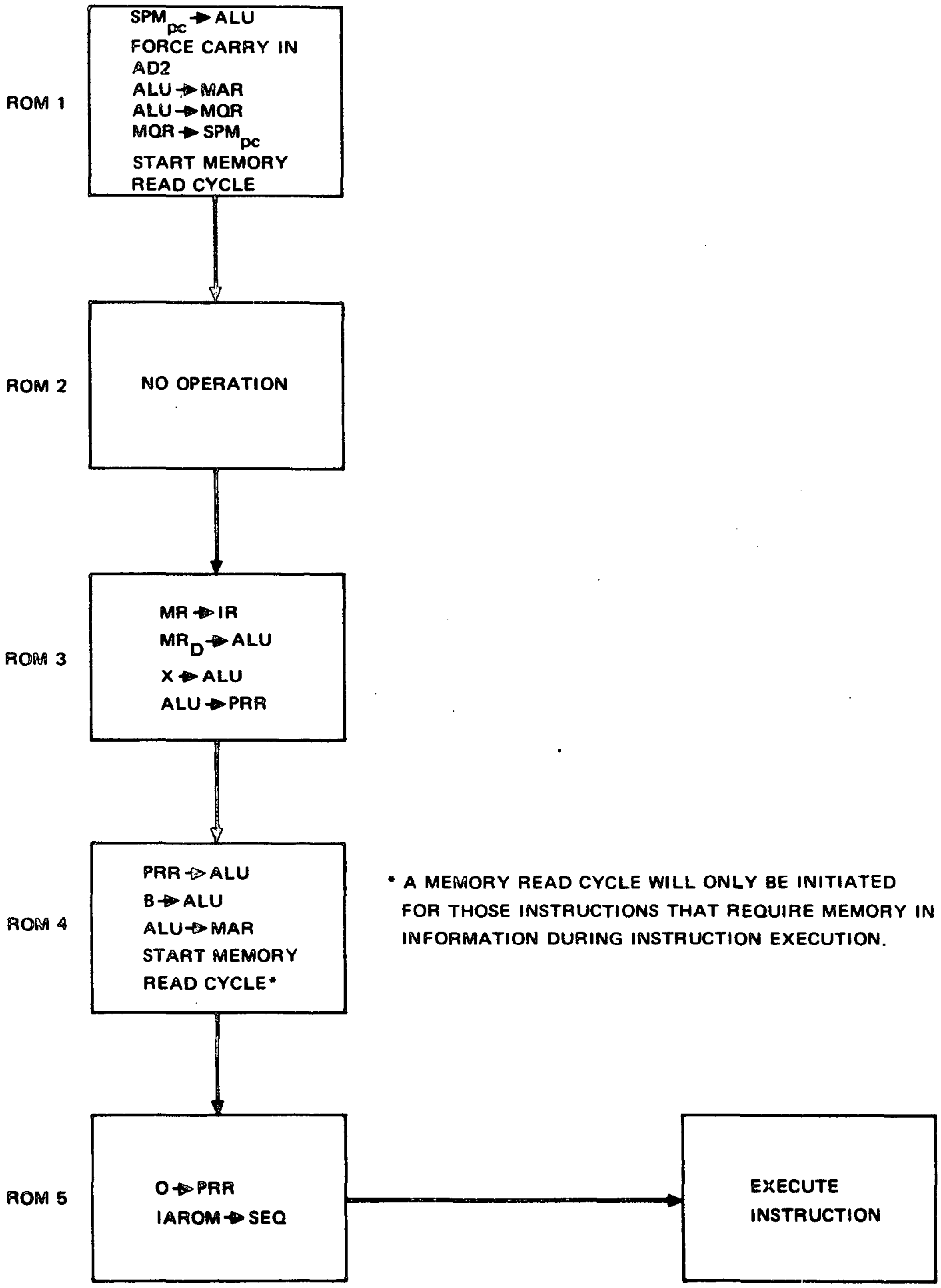


Figure 2-9. Control Unit (CU) Flow Diagram



* A MEMORY READ CYCLE WILL ONLY BE INITIATED FOR THOSE INSTRUCTIONS THAT REQUIRE MEMORY IN INFORMATION DURING INSTRUCTION EXECUTION.

Figure 2–10. Fetch Cycle Flow Chart

controlled by a bit of IAROM as previously mentioned. Figure 2-11 shows the format of the MROM words. Use of 72 of the 80 bits is shown. Additional bits will be used for the microprocessor configuration.

Configuration Control Unit (CCU)

The CCU is the most significant element of hardware which must be added to simplex computers to make a multiprocessor. The CCU has three main parts: address translation, communication control, and state control. All hardware elements have a CCU; however, the main memory modules do not have address translation, and the input/output controllers have no preferential storage register.

Address Translation Unit (ATU)—Programs are written without regard to the physical memory module where they will be loaded. The program address, called a logical address (Z_L), is the sum of the displacement from the instruction word and the base and index registers if they are called out by the instruction word; i. e. $Z_L = D + (X_i) + (B_j)$. The ATU replaces the memory module designation part of the logical address with the physical address of the module containing the information. Additionally, a special memory reference is provided for a preferential storage area (PSA). Here not only the physical module but a part of the address within the module is provided by the ATU. Figure 2-12 shows the format of the addresses before and after translation.

An interconnection diagram of the ATU to a CPU is shown in Figure 2-13. This diagram shows how the existing SUMC data paths are used for loading and storing the address translation registers (ATR). Figure 2-14 shows a working-level block diagram of the ATU for a CPU. The logical address Z_L comes into the ATU, its module designation is decoded, and if nonzero is used to gate the contents of a four-bit register (in either ATR-1 or ATR-2) into a physical module select decoder. If the decoded logical module designation is zero, the PSA module is selected, and the PSA value of K is used instead of the logical value of K. As an optional feature, a primary and alternate memory module can be designated as PSA. The same starting address would be used within both modules.

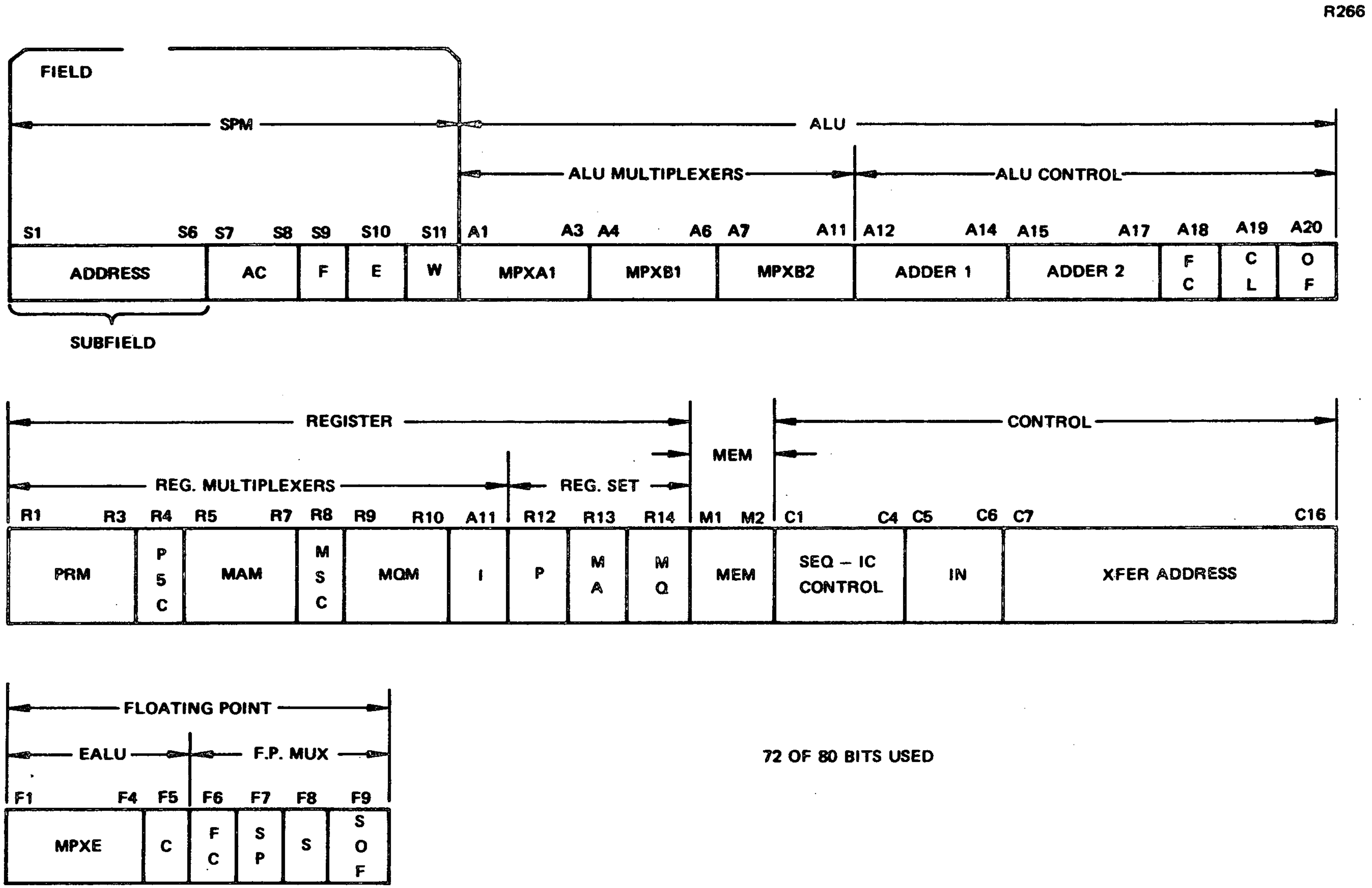


Figure 2–11. Microinstruction Word Format for the Microprogrammed Read Only Memory (MROM)

ADDRESS BITS (32)		UNUSED	MODULE (4)	SUB MODULE (13-14)	
				K	L
NON-PSA REFERENCE	GENERATED STORAGE ADDRESS (LOGICAL)	ZEROS	4 BITS NON-ZERO (LOGICAL)	ADDRESS WITHIN MM MODULE (LOGICAL AND PHYSICAL ARE THE SAME)	
	TRANSLATED STORAGE ADDRESS (PHYSICAL)		TRANSLATED 4 BITS DEFINE PHYSICAL MM MODULE 1 TO15		
PSA REFERENCE	GENERATED PSA REFERENCE (LOGICAL)	ZEROS	ZEROS	ZEROS	LOGICAL WITHIN PSA BLOCK
	TRANSLATED PSA REFERENCE (PHYSICAL)		PRESTORED 4 BITS DEFINE PHYSICAL MM MODULE 1 TO15	PRESTORED BITS DEFINE START ADDRESS OF PSA BLOCK	LOGICAL WITHIN PSA BLOCK
				LOGICAL AND PHYSICAL ADDRESS WITHIN MM MODULE	

NOTE: L is the address within preferential storage and K is part of the PSA address provided by the ATU.

Figure 2-12. Address Translation Formats

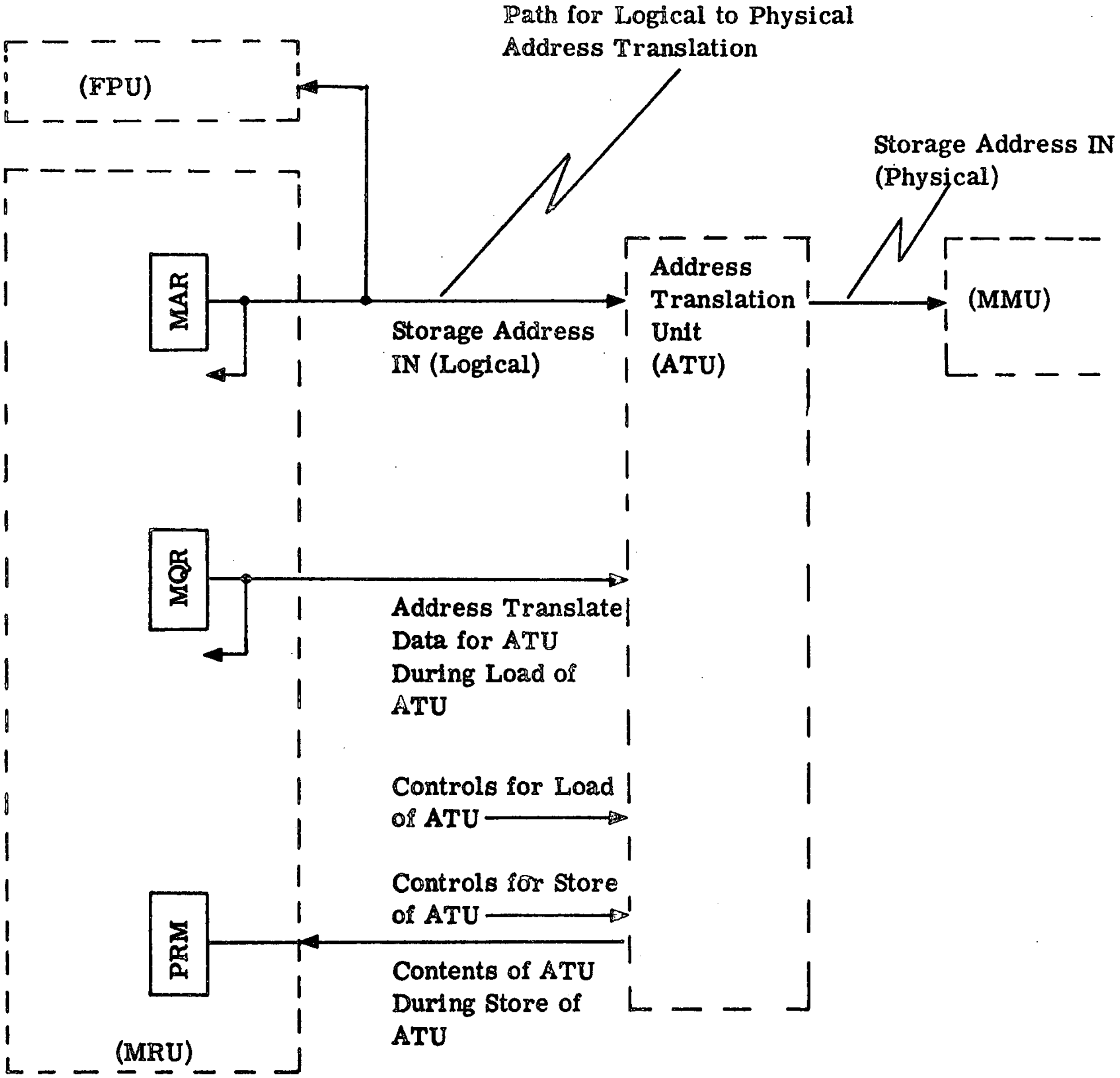

Path for Logical to Physical Address Translation
(FPU)
Storage Address IN (Physical)
MAR
Address Translation Unit (ATU)
(MMU)
Storage Address IN (Logical)
MQR
Address Translate Data for ATU During Load of ATU
Controls for Load of ATU
Controls for Store of ATU
PRM
Contents of ATU During Store of ATU
(MRU)

Figure 2-13. ATU/CPU Interconnection

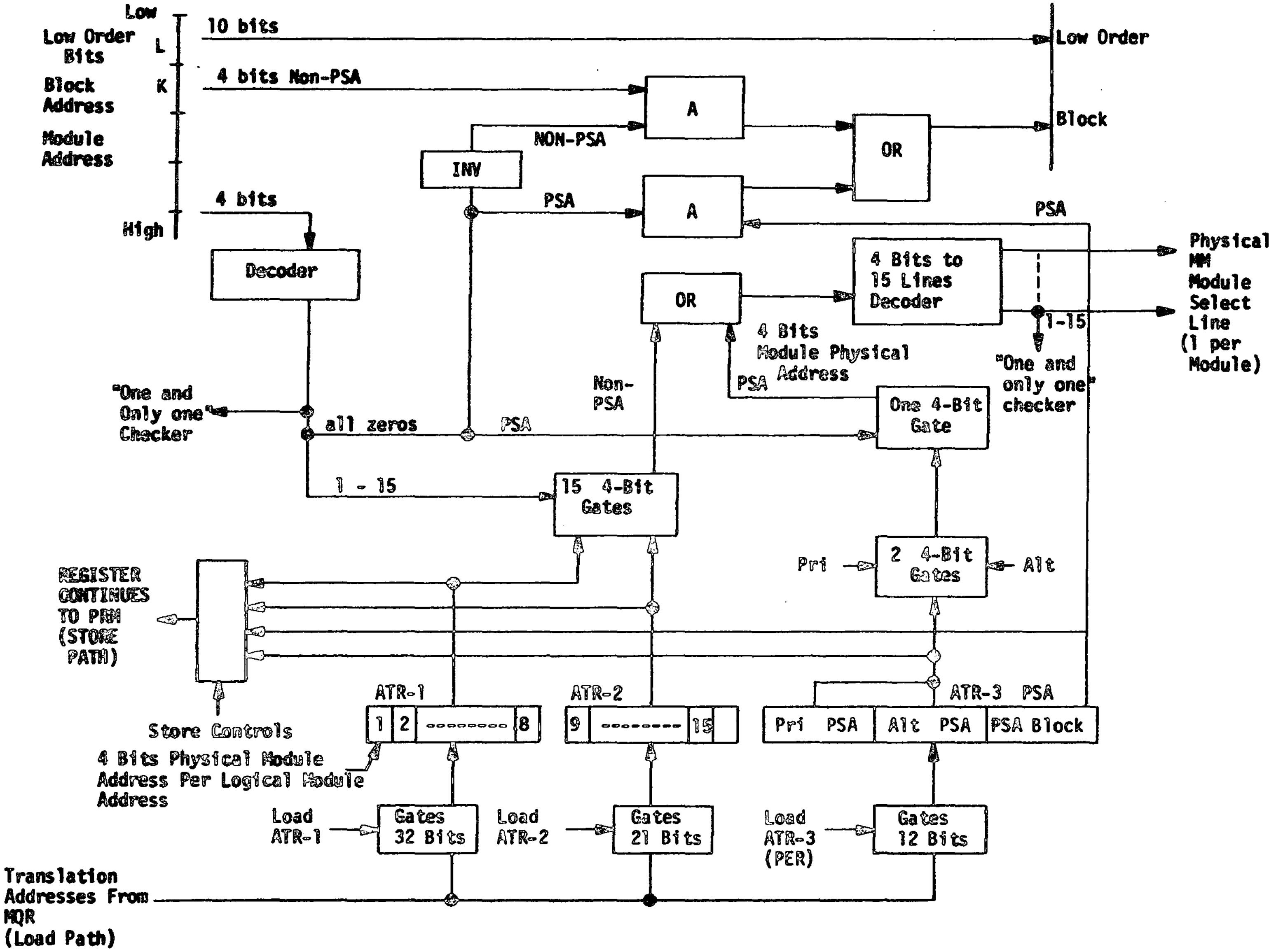


Figure 2-14. Address Translation Unit Block Diagram

Figure 2-15 shows the interconnection of the ATU for an IOC. This ATU is like that of a CPU, except that no PSA is used, and load and store ATU is as shown (with memory).

Communication Control—Each functional element of the multiprocessor has communication control logic as part of its configuration control unit (CCU). This logic uses the communication state register (CSR) to enable/inhibit communication with other elements of the system. This feature provides the ability to isolate hardware elements from the multiprocessor and provides redundancy in setting up communications with memory in the multiprocessor configuration; i. e., for a CPU or IOC to be able to read or write in a memory module, the ATR of the CPU or IOC must contain the physical module address of the MM, the CPU or IOC's CCR must be enabled to communicate with the MM, and the MM must be enabled to communicate with that CPU or IOC. Thus, inadvertent altering (or using) of MM contents due to software errors or hardware faults is minimized.

A listing follows which shows the communication control between units. It can be seen that the CPU can control transmitting and receiving from other CPU's and IOC's independently, for added flexibility.

A. Definition of Communications for CPU:

15 bits	for	MM modules
4 bits	for	transmit to CPU control
4 bits	for	Receive from CPU control
3 bits	for	transmit to IOC control
3 bits	for	receive from IOC control
29 bits		

B. Definition of Communications for MM:

4 bits	for	intercommunications with CPU
3 bits	for	intercommunications with IOC
7 bits		

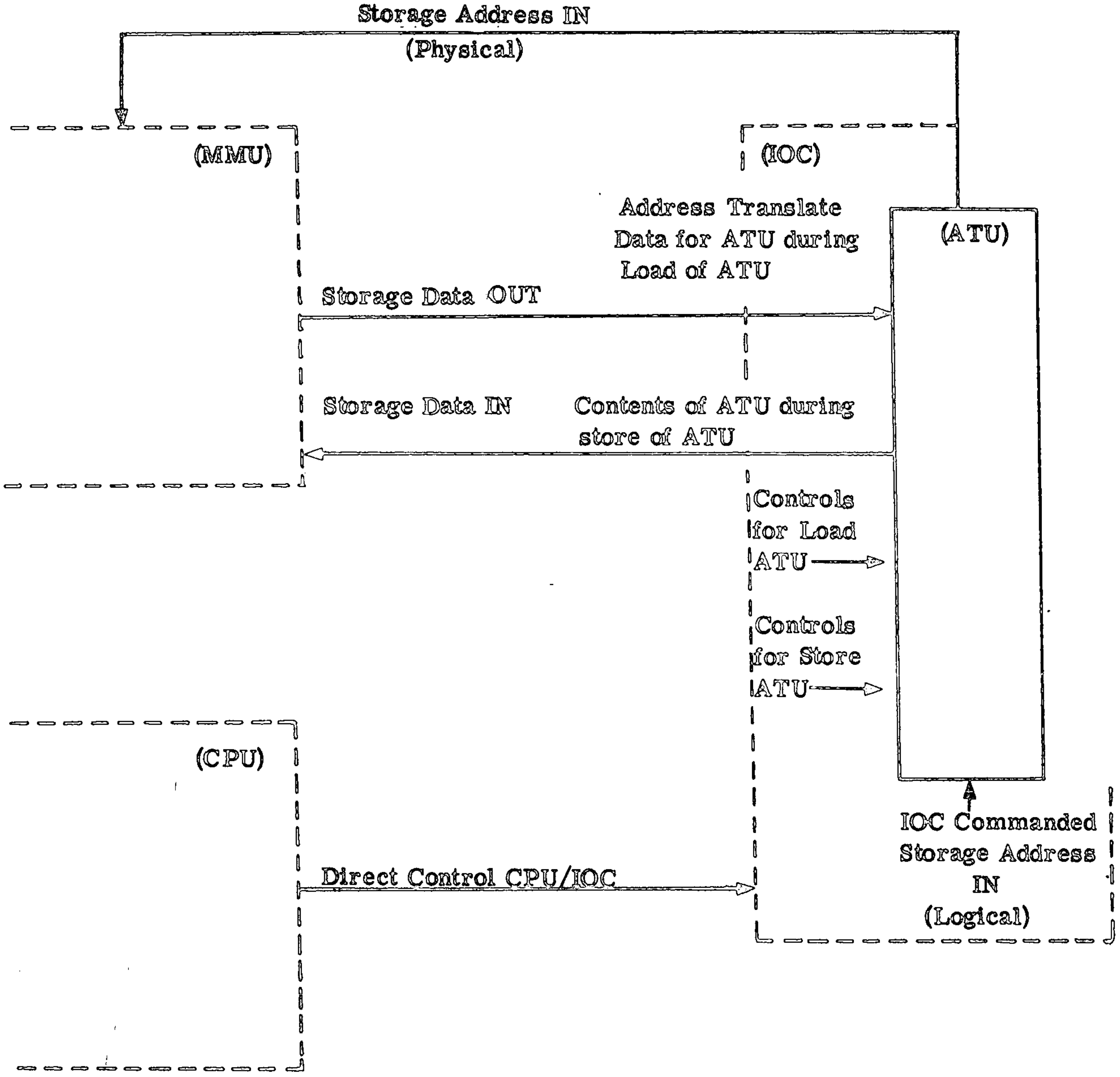


Figure 2-15. ATU/IOC Interconnection

C. Definition of Communications for IOC:

15 bits	for	MM modules
4 bits	for	Intercommunications with CPU
19 bits		

State Control—All states and their options are provided in the CPU's. State 3 is not applicable to MM's or IOC's. When a unit enters state 1, it will complete its present operation then halt:

A. CPU—Complete current instruction but do not initiate next fetch cycle. Hold until operational state is reestablished or unit is turned OFF.

B. MMU—Complete present memory cycle; then hold BUSY line up to inhibit further use.

C. IOC—Complete transfer with device now addressed but do not initiate communication with new device.

The CSR is used for state control as well as communication control. The following table shows use of three bits for state control. A single bit is reserved for state 3, and this bit is used only by the CPU's CSR.

State	CSR Code
0	0 0 0
1	0 0 1
2	0 1 0
3	1 1 0

Complete CSR usage is shown in the following table.

Unit	Communication	State	Total
CPU	29	3	32 bits
MMU	7	2	9 bits
IOC	19	2	21 bits

If any CPU goes to state 3, the following conditions apply:

A. All other CPU's will stop at the end of the present MROM cycle.

B. State 3 for all other CPU's will be inhibited.

C. All IOC's will be stopped immediately. Incoming data, if any, will not be processed, and no data will be transmitted, even if expected by a device.

D. The state 3 signal will not alter the contents of the CCU's for other elements, but will enable a LOAD CCU EXTERNAL instruction for all elements.

E. A state 3 CPU can use any memory module not manually isolated from the system.

State 3 is terminated as follows:

A. The state 3 CPU executes a Load CCU Self to return to state 2.

B. Removal of the state 3 signal from the other CPU's will allow them to operate according to their present CCU contents. If the CPU had its CCU loaded during the reconfiguration, it will start a new instruction fetch cycle. If the CPU was not altered in reconfiguration, it will continue executing its MROM as if never halted.

C. IOC's will operate according to their present CCU, assuming that they have been properly initialized during reconfiguration to unscramble any I/O operations which were affected by the state 3 halt.

CCU Loading—Two instructions are provided to load the registers in the CCU's: Load CCU Self (LCCU/S) and Load CCU External (LCCU/E). The LCCU/S instruction is used by any CPU to load its own CCU, whereas the LCCU/E is used for changing the configuration of another CPU, an IOC, or a MMU.

Existing data paths are used to the maximum extent in reconfiguration; therefore, data for loading external CCU's comes from MMU's on the memory buses. During LCCU/E, the CCU of the state 3 CPU participates in memory addressing in the normal way. Performing LCCU/S cannot simultaneously use the ATR's and change them so data are taken from SPM for the self load.

Other features of LCCU/E are as follows:

A. LCCU/S does not require state 3.

B. LCCU/E requires state 3.

C. For LCCU/S, the contents of SPM goes via the MOR to the CCU registers. Five bits of the D field of the instruction word selectively inhibit loading of ATR-1, ATR-2, ATR-3 (PSR), and the state and communications portions of the CSR. The A field of the instruction word identifies the first of the four SPM locations used to load the four CCU registers.

D. For LCCU/E, main memory is addressed normally (Z_L = D + X + B) to obtain the data for loading the four CCU registers. The CPU, IOC, or MMU is selected by 1 of 22 bits in SPM addressed by the A field of the instruction word. Five other bits of the same SPM word are used to selectively inhibit loading of AT-1, AT-2, etc., the same as the D field is used for LCCU/S.

The LCCU instructions and other instructions associated with the CCU are shown in Figure 2-16.

Storage Protection

Three features are provided to help eliminate potential impact of one CPU on another by simultaneous use of the same program or a CPU using the wrong storage area: read protection, write protection, and subroutine lockout.

Every time memory is addressed, the most significant bits of the address are used to access a six-bit word contained in a separate small section of memory. The six bits include one bit for read protect, one for write protect and a four-bit key word, which functions as follows:

A. If the read bit is a 1, the CPU supplied key must match the stored key or the memory will not send the data to the CPU and a key violation signal will be sent to the CPU.

B. If the write bit is a 1, the keys must match or the CPU data will not be stored and a violation signal will be sent to the CPU.

C. If neither bit is a 1, all requests for memory service will be honored.

(1)	LCCU	=	Load CCU/CPU's Self (Does not require State 3)				
	OP Code	-	A	-	D	+	Microaction
	LCCU	-	Modifies SPM Subfield to access: o ATR-1 o ATR-2 o PSR o CSR		Inhibits Load of: o ATR-1 o ATR-2 o PSR o State o Communications		SPM→MQR→CCU for each of: o ATR-1 o ATR-2 o PSR o CSR
(2)	LCCU/E	=	Load CCU/C (Requires State 3)				
	OP Code	-	A		B + X + D→Z	+	Microaction
	LCCU/E	-	Modifies SPM Subfield to access: Select Control →MQR. Selects: o 3 IOCs o 4 CPUs o 15 MMs Inhibits Load of: o ATR-1 o ATR-2 o PSR o State o Communications (Can not select CPU Self).				Access MM per Z in MAR + Direct Control: Gates per Micro: o ATR-1 o ATR-2 o PSR o CSR Selects per MQR: o IOC o CPU o MM Inhibits per MQR: o ATR-1 o ATR-2 o PSR o State o Communications Reconfiguration Order per Op Code
(3)	SCCU	=	Store CCU/CPU's Self (Does not require State 3) and store CCU/E (Requires State 3)				
	Op Code	-	A		B + X + D		
	SCCU	-	Modifies SPM Subfield to access: Select Control→MQR Selects: o 3 IOCs o 4 CPUs o 15 MMs (Can select CPU self) One Element Selected per each instruction.		B + X + D→ Z	+	Microactions CCU to Storage Data IN to MM per Z in MAR + Direct Control Gates per Micro: o 4 Regs. Selects per MQR: o IOC o CPU o MM Reconfiguration Order per Op Code

(4) Go to State 3 (Requires State 2 or 3)
Microaction 1 Set Bit in CSR of own CPU to indicate State 3.

(5) Power Down (Requires State 3)
via Direct Control to a State 1 Element.

(6) Fetch after Stopped State Automatic after Stopped State
Microaction 1 0→MAR
Start Memory Read and fetch from Logical 0 of PSA
Transfer to ROM2 to Fetch.

(7) Power Up (Requires State 3)
via Direct Control.

Figure 2-16. Configuration Control Instructions

The size of memory block which is protected is a trade between potential storage inefficiency (too large a block) and requiring a large storage for the keys (too small a block). The block size will likely be in the range of 128 to 512 words. Thus, an MMU of 8, 192 words would require a key storage area of 16 to 64 six-bit words.

Storing of protection keys in memory will be by a special instruction— "Establish Protection Word"—which will be a privileged instruction. Four bits of the PSR will hold the protect key for the CPU. A "Load Protect Key" instruction will be provided for loading the key in the PSR and a "Store Protect Key" will allow storing the data in memory or SPM.

To prevent more than one CPU from executing such programs as the executive control program at the same time, a "Test and Set" instruction is provided, which performs as follows:

A. A "protected program" has a storage location two words ahead of it which contains zeros or a "busy" code.

B. A Test and Set instruction reads the contents of this storage location and, if it is a busy code, the CPU is notified by incrementing the instruction counter by one (to a location containing a jump to the busy routine). If the contents of the tested location is not a busy code, a busy code is stored in that location. Then the instruction counter is incremented by two to the beginning of the protected routine.

C. If a Test and Set instruction tests a not-busy location, the MMU busy signal will not be dropped until after setting the location to the busy code.

D. The test location can be cleared by a conventional store signal.

To provide flexibility in reacting to a busy signal, a "Delay" instruction is provided to allow the programmer to insert a delay (count n MROM cycle times) if he wishes. Of course, the "Busy Routine" could also branch to other functions before trying the protected program again.

Executive Control

The proposed hardware features allow considerable flexibility in creating a powerful Executive Control Program (ECP). Some of the salient features are summarized here. A detailed description is given in Appendix A of this report.

A. Direct CPU to CPU and CPU to IOC communications augment communication through storage to provide redundancy and "instantaneous" access.

B. Any CPU can execute the ECP.

C. Through state control and the ECP, any CPU can remove a malfunctioning CPU from the system.

D. Through state control and special instructions, the ECP can power up and down any unit.

E. Special instructions are included to initialize I/O.

F. Upon task completion, each CPU returns to the ECP for a new assignment.

G. All configuration control can be done by the ECP.

To facilitate a CPU identifying itself, a Store Identity instruction is included. This instruction generates a signal via MROM decoding which forces one of four bits in the PRR. The contents of PRR is subsequently stored using normal addressing. Installation wiring determines which bit of the PRR is set. Any CPU plugged into the location identified for CPU 1 will have bit one set, etc.

Preferential Storage

An area within main storage is identified as the preferential storage area (PSA) for that CPU. Preferential storage is provided to allow simple logical addressing to selected critical program linkages, status data, restart data, etc. The PSA has the following features and characteristics:

A. Any logical address with zeros for the module number is identified as a PSA address.

B. In PSA addressing, the most significant k bits of the address come from the PSR, and the least significant m bits come from normal logical address $Z_L = D + X + B$.

C. If a PSA address overflows the PSA size, an invalid address indication is given by the ATU.

D. The PSA can be addressed normally using a full logical address translated through ATR-1 or -2.

E. As an option, the PSA can use special instructions such as Load or Store PSA with bits from the instruction word used directly to modify the logical address. This would make the PSA seem like several small tables rather than one large one.

F. An option has also been described for having an alternate PSA located at the same relative location within a different MMU.

Reconfiguration State Monitor

The reconfiguration state (state 3) should be monitored separately from CPU operations. When reconfiguration is required and "state 3" is entered, it is good design that an external (hardware) monitor supervise timely entry into and exit from state 3 independently of "operator" type supervision. The functions assigned to the monitor should include the following:

A. Verification that state 3 did occur when expected.

B. Verification that state 3 occurred within a preprogrammed time interval.

C. Verification that the system was configured into a minimum viable configuration.

D. Signal "ALARM" upon detection of improper operation.

The executive program knows when it needs to reconfigure. It is reasonable to expect that the reconfiguration executive program can estimate the expected duration of the reconfiguration process. In those areas where the duration is uncertain, a reasonably higher duration interval would be specified. As a part of programming precautions, the entry into the loop to "Go to state 3" would be preceded by an "enable" function. The instruction "Enable" signals the monitor that a reconfiguration will follow, thereby setting the stage for the operation.

The reconfiguration state monitor unit (RSMU) is shown as a part of the multiprocessor system in Figure 2-17 and as a unit block diagram in Figure 2-18.

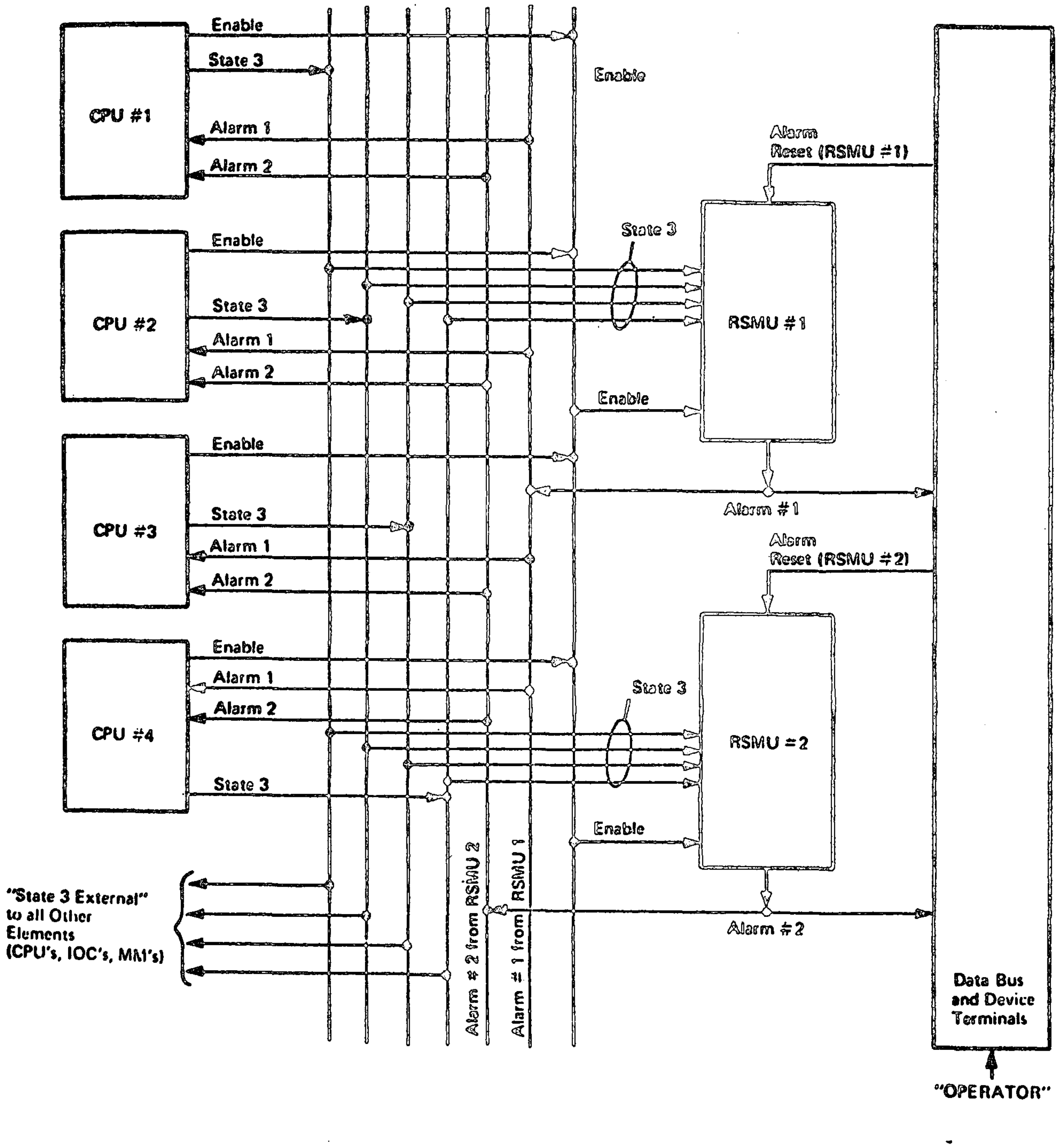


Figure 2-17. RSMU System Diagram

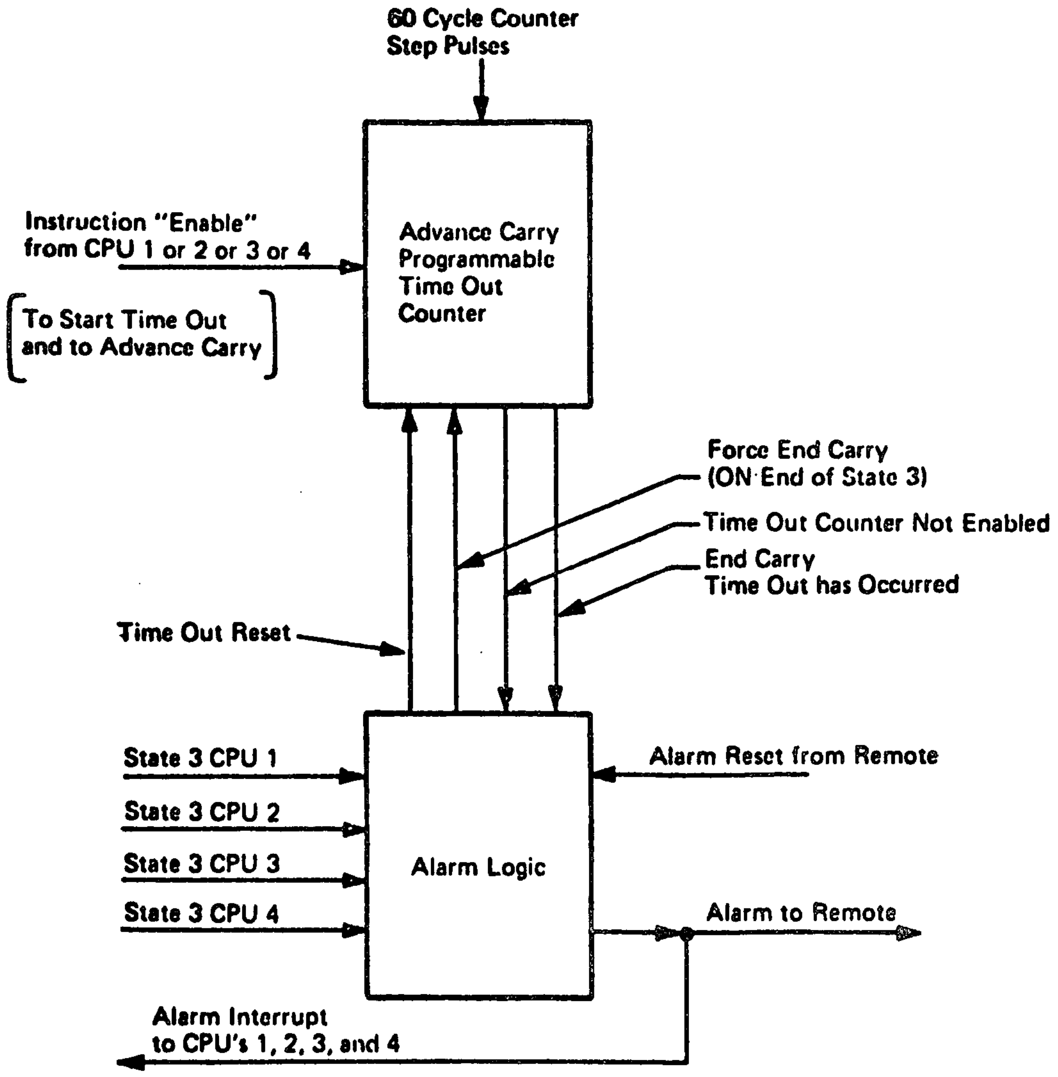


Figure 2-18. RSMU Block Diagram

Reconfiguration may be executed in program "stages." Each stage would be separately and independently monitored. The executive program first would "enable" for reconfiguration. The instruction Enable is used to start a binary timeout counter in the RSMU. The timeout is used to define the reconfiguration period. The RSMU will check that the reconfiguration was begun and was completed in the timeout period, and, if not, would signal Alarm.

As the reconfiguration period is a variable depending on the amount of configuration definition to be changed, it is efficient to be able to specify different timeout periods consistent with the work to be done. To provide for a variable time multiple use is made of the instruction Enable line. Upon first execute, the Enable starts the RSMU counter. Upon subsequent executes, the Enable advances the high-order carry of the binary counter so as to halve the timeout. This is shown logically on Figure 2-19. Additional timing details of the variable programmed timeout is shown in Figure 2-20.

Having provided a monitor, and having provided an instruction to start and set a variable timeout, it now remains to state the conditions to be supervised. The Alarm logic is shown on Figure 2-21.

The Alarm logic looks at the state 3 lines from each CPU. If state 3 behaves as expected, Alarm will not be signalled. If state 3 does not behave as expected, Alarm is signalled. If the reconfiguration is signalled as complete, the timeout condition is forced to early completion, thereby not requiring a wait through the preset timeout period and thereby ending the monitor operations for the one period, allowing for immediate readiness of the monitor for the next reconfiguration period. The timeout period is forced to early completion when the state 3 condition is ended.

In addition to checking that state 3-behaved as expected, the Alarm logic can be expanded to include a check that at time of expiration of timeout the system was configured into a minimum acceptable viable state, such as a minimum of two CPU's in state 2. The RSMU can be designed to detect all

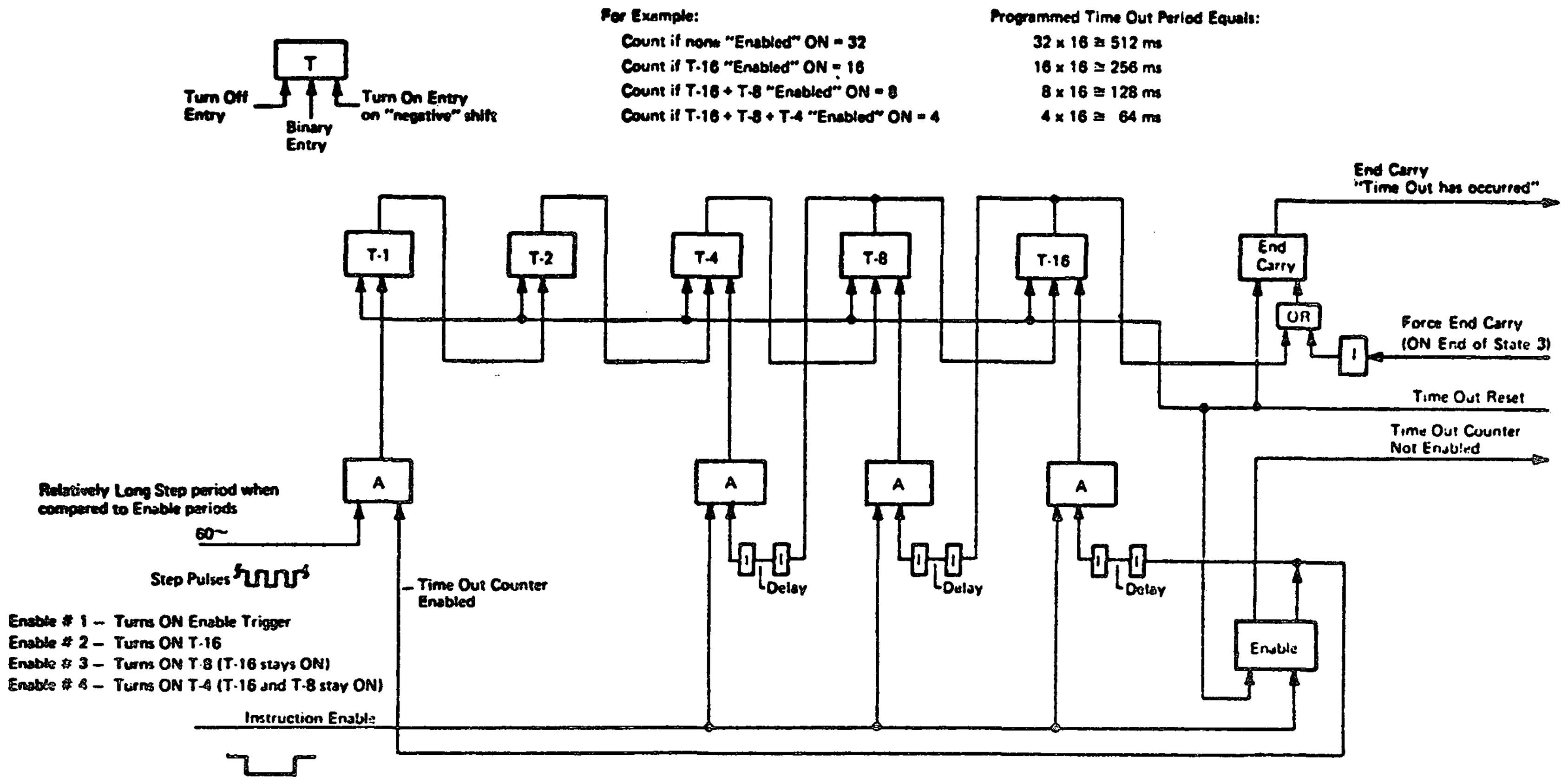

T
Turn Off Entry
Binary Entry
Turn On Entry on "negative" shift
For Example:
Count if none "Enabled" ON = 32
Count if T-16 "Enabled" ON = 16
Count if T-16 + T-8 "Enabled" ON = 8
Count if T-16 + T-8 + T-4 "Enabled" ON = 4
Programmed Time Out Period Equals:
32 x 16 ≅ 512 ms
16 x 16 ≅ 256 ms
8 x 16 ≅ 128 ms
4 x 16 ≅ 64 ms
T-1
T-2
T-4
T-8
T-16
End Carry
End Carry
"Time Out has occurred"
OR
I
Force End Carry
(ON End of State 3)
Time Out Reset
Time Out Counter
Not Enabled
A
A
A
A
Delay
Delay
Delay
Enable
Relatively Long Step period when compared to Enable periods
60~
Step Pulses
Time Out Counter Enabled
Enable # 1 – Turns ON Enable Trigger
Enable # 2 – Turns ON T-16
Enable # 3 – Turns ON T-8 (T-16 stays ON)
Enable # 4 – Turns ON T-4 (T-16 and T-8 stay ON)
Instruction Enable

Figure 2-19. Time Out Counter

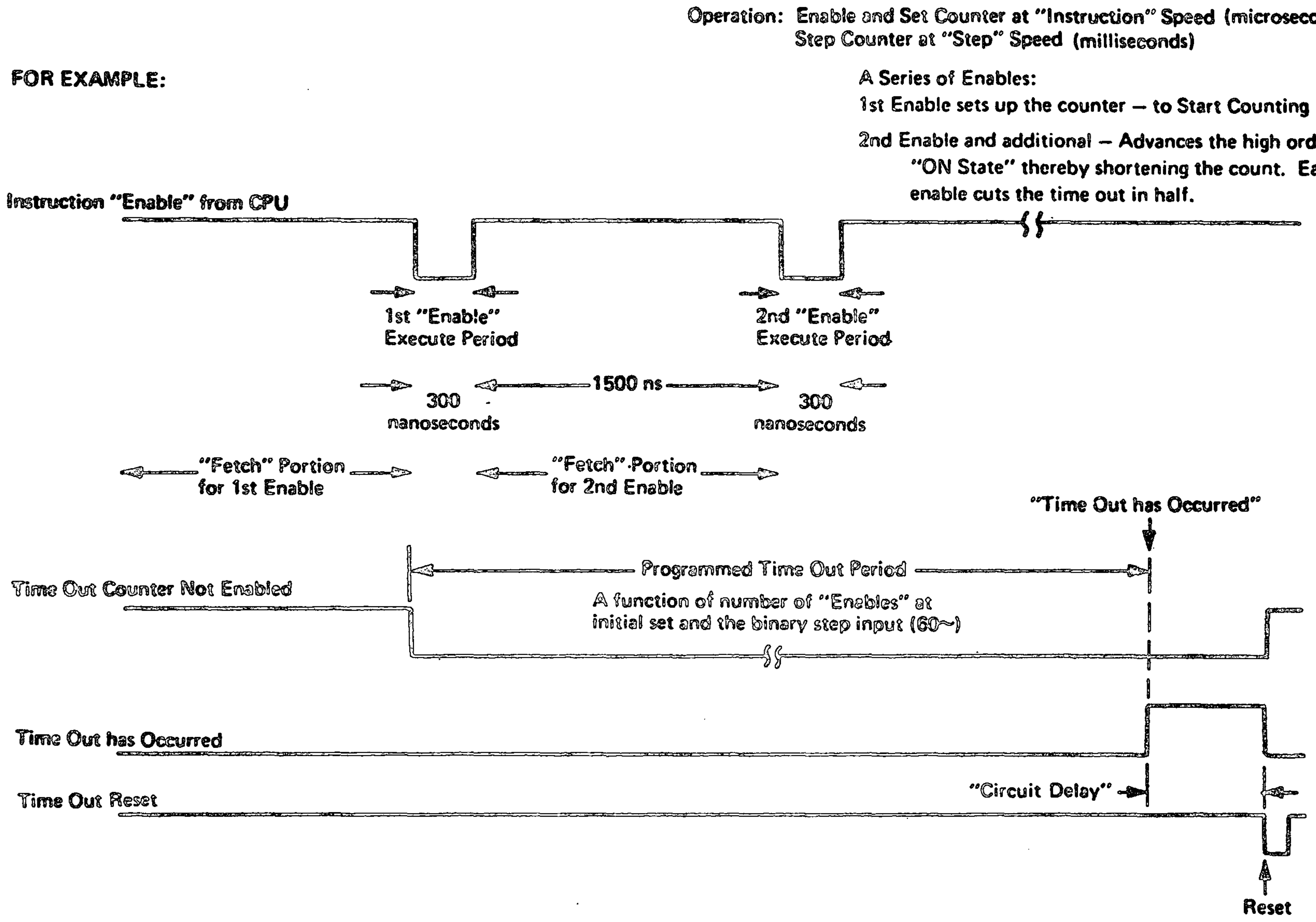

Operation: Enable and Set Counter at "Instruction" Speed (microseconds)
Step Counter at "Step" Speed (milliseconds)
FOR EXAMPLE:
A Series of Enables:
1st Enable sets up the counter – to Start Counting
2nd Enable and additional – Advances the high order "ON State" thereby shortening the count. Each enable cuts the time out in half.
Instruction "Enable" from CPU
1st "Enable" Execute Period
2nd "Enable" Execute Period
300 nanoseconds
1500 ns
300 nanoseconds
"Fetch" Portion for 1st Enable
"Fetch" Portion for 2nd Enable
"Time Out has Occurred"
Programmed Time Out Period
A function of number of "Enables" at initial set and the binary step input (60~)
Time Out Counter Not Enabled
Time Out has Occurred
Time Out Reset
"Circuit Delay"
Reset

Figure 2-20. RSMU Timing Diagram

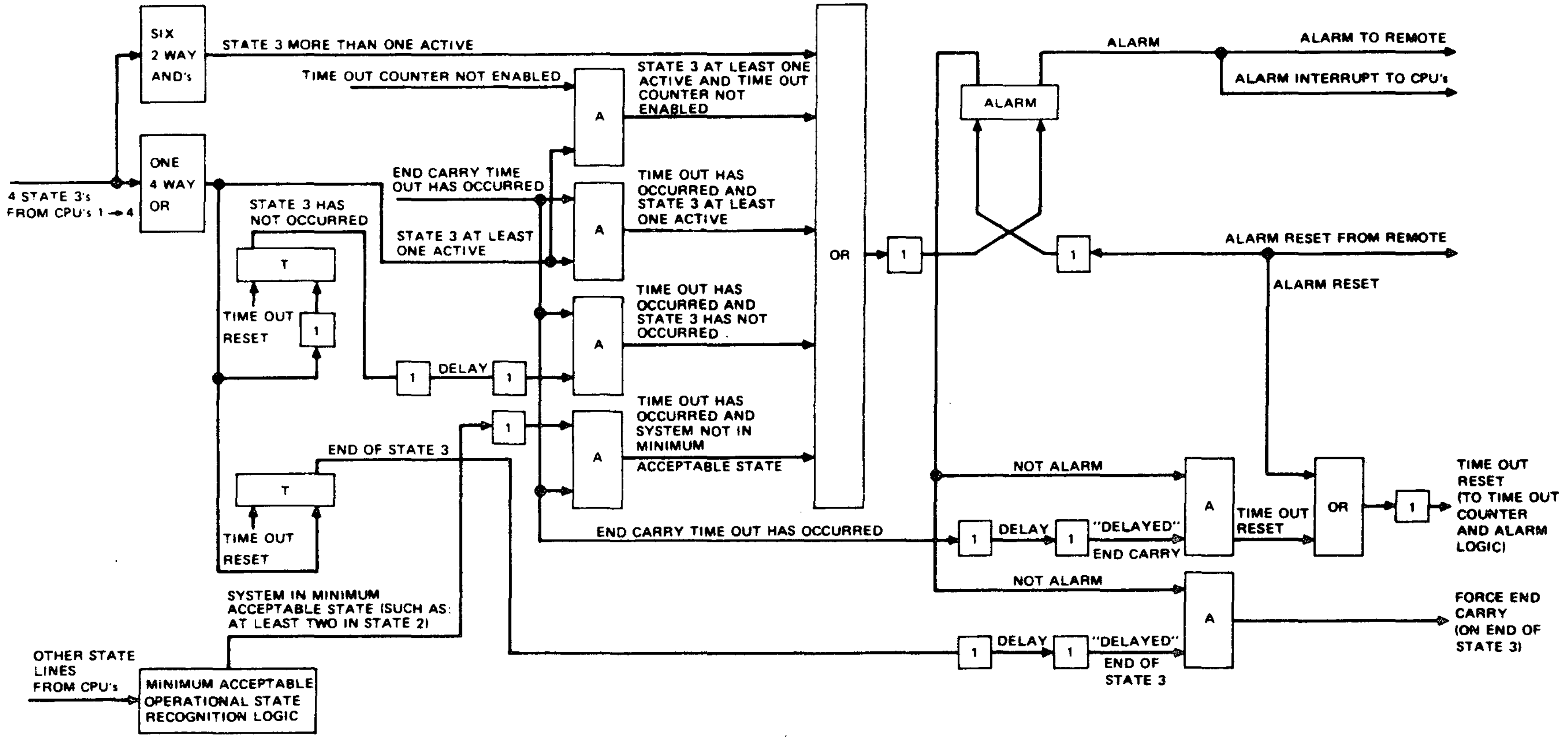


Figure 2–21. RSMU Alarm Logic

reasonable state associated logic constent with keeping the RSMU simple to enhance its reliability.

The importance of the state 3 monitor function requires that all reasonable care be taken to assure reliable detection and alarm. A duplexed RSMU is, therefore, recommended. Either RSMU can signal Alarm. The two RSMU's work in parallel, but independently. Duplexed RSMU's, periodically exercised and checked for operability under CPU control, provide reasonable assurance of reliable monitoring, detection, and alarm. By allowing each to be separately locked in Alarm state, the other can be uniquely tested for operability of all its modes.

The Alarm from the RSMU is signalled both to the remote world and to the CPU. If the CPU is well, the Alarm Interrupt when masked ON will provide opportunity for CPU-directed reset and recovery. The Alarm can only be reset through the external world, either by the CPU through the Data Bus to Device Terminal link or by an operator, either local or remote control.

Should the multiprocessor system desire, it can ignore the Alarm, but the Alarm signal to the external world would continue.

If CPU's become stuck in state 3, the Initial Load Reset (ILR) function is a necessity to bootstrap the system into operability.

If CPU's are not stuck in state 3, but operating incorrectly, then it is possible for the Alarm interrupt to the CPU to call in needed software to get the system into a more viable state. If the CPU is well, it will have the option of turning off the Alarm via the data bus. If the CPU cannot get through to the Alarm to reset it, then the Alarm signal remains on until an operator turns it OFF.

An initial load (bootstrap) should be provided to allow startup with all MMU's and SPM's either blank or with unknown contents. The Initial Load

Sequence (ILS) is preceded by a master clear signal which zeros all CPU registers, with the following exceptions.

A. One CPU is placed in state 3.

B. All other CPU's are placed in state 2.

C. All communications are enabled.

ILS is started by external generation of one of four initial load signals ILS-1 through -4. The number of the CPU placed in state 3 corresponds to the number of the ILS signal. The ILS forces an ILS op code and address zero. Via MROM, the ILS instruction loads a short bootstrap starting at address zero. Upon completion of the bootstrap load, the program counter is set to address zero and normal processing starts. The bootstrap loaded now initiates a memory load from a mass memory until a special end of block signal is detected. Control is then transferred to a startup routine via an interrupt, and normal software takes over.

2. 1. 4 SUMC Development Requirements

This section summarizes recommended changes and associated rationale for extension of the basic SUMC capabilities to multiprocessor and specialized input/output operations. It includes interrupt processing, storage protection, and privileged instruction features which may be incorporated in either simplex or multiprocessor configurations. References are included to other sections of the report for more detailed explanation of the changes recommended.

2. 1. 4. 1 Multiprocessor Operations

The basic requirement for a multiprocessor in the Space Station Program is the ability to configure the system elements to meet the particular computation requirements or to overcome malfunctions when they occur. The approach outlined in the following subparagraphs summarizes additions and changes recommended for multiprocessor operations. Further details are contained in Section 2. 1. 3. This approach was selected because it requires only a small amount of additional hardware and permits a high degree of flexibility in configuring an operational system.

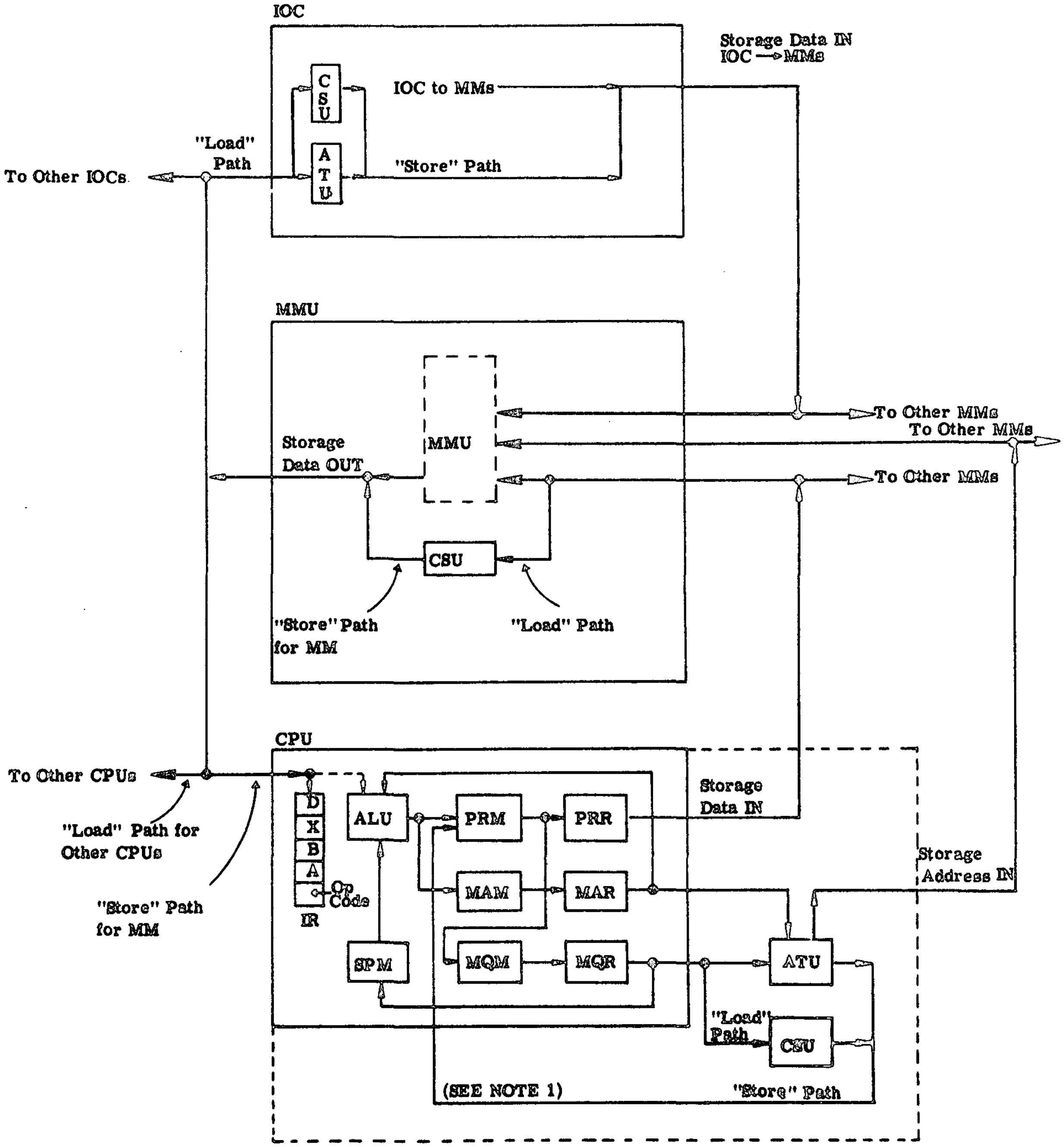
Configuration Control

The approach to configuration control is to provide configuration control units associated with each CPU, IOC, and main memory module in the system. These units control the permissible communications paths between elements, control the state of each element in the system, and translate from a logical address to a physical address in main memory in order to facilitate the configuration control process. These units involve logic which may be either physically located within each system element or mounted separately on an interconnecting assembly. The configuration control units for the CPU's and IOC's consist of communications and state control registers and address translation units. The main memory module configuration control units contain only the communications and state control registers.

Communications Control—Permissible communications paths are established under executive program control through the use of communication control registers and input/output gates associated with each element in the system. The setting of communication control register bits controls input/output gates between CPU's and IOC's, CPU's and main memory modules, and IOC's and main memory modules. Loading and storing of the communications control registers is illustrated in Figure 2-22. After loading, the contents of the communications control registers are read into a preferential storage area in main memory for verification and reference purposes.

State Control—The state of each element in the system is established by any CPU which is directed under executive program control to configure or reconfigure the system. Each element has an associated state control register which is decoded to determine the state of that element. The states include maintenance, standby, operational, and a special reconfiguration state used only by CPU's to establish the configuration of the system. The state registers are loaded and stored similarly to that of the communication control registers described above.

Address Translation—The address translation feature allows the reassignment of memory locations between memory modules when the system is reconfigured. This allows one memory module to be substituted for another



NOTE 1. It appears feasible and more desirable to go through the ALU to get to PRR by using an existing port like the Input I/O at MPXB1 rather than a new port at PRM.

Figure 2-22. Configuration Control System Diagram

without extensive software modifications. For example, if one memory module is malfunctioning or being maintained, another memory module in the system may be substituted by changing the module translation address.

Implementation of the address translation feature is illustrated in Figure 2-23. Two address translation registers (ATR-1 and ATR-2) contain fifteen sets of four high-order bits defining the translated memory module address (one set per module). The logical address defined by the computer program is translated to the physical address by referring to the translated module address stored in the address translation registers. The translated module address is then decoded to select the particular memory module to be accessed. A preferential storage register (ATR-3) indicating the start of the physical preferential storage area associated with each CPU and IOC may be identified. The preferential storage area is indicated when the four high-order bits of the logical address are all zeros. Loading and storing the contents of the address translation registers is similar to that described under configuration control.

Multiprocessing Instructions

Instructions necessary to control the configuration of the multiprocessing system are illustrated in Figure 2-16. Details of the multiprocessing instructions are contained in Sections 2. 1. 3 and 2. 3 of this document. These instructions are required for the loading and storing of the configuration control registers and controlling the state of elements within the system. The LCCU and SCCU instructions are required to load and store the CCU from its own associated CPU. The LCCU/E and SCCU/E instructions are required to load all other CCUs from the CPU controlling the configuration process. The "Fetch after Stopped State" is an automatic instruction which starts with the logical zero of each CPU's preferential storage area to initiate CPU operations under the new configuration. Instructions to power up or power down system elements are accomplished through direct control to the element involved while the system is in the reconfiguration state.

A Test and Set instruction is required to determine whether or not particular blocks of storage are being used by another CPU, and thereby are not

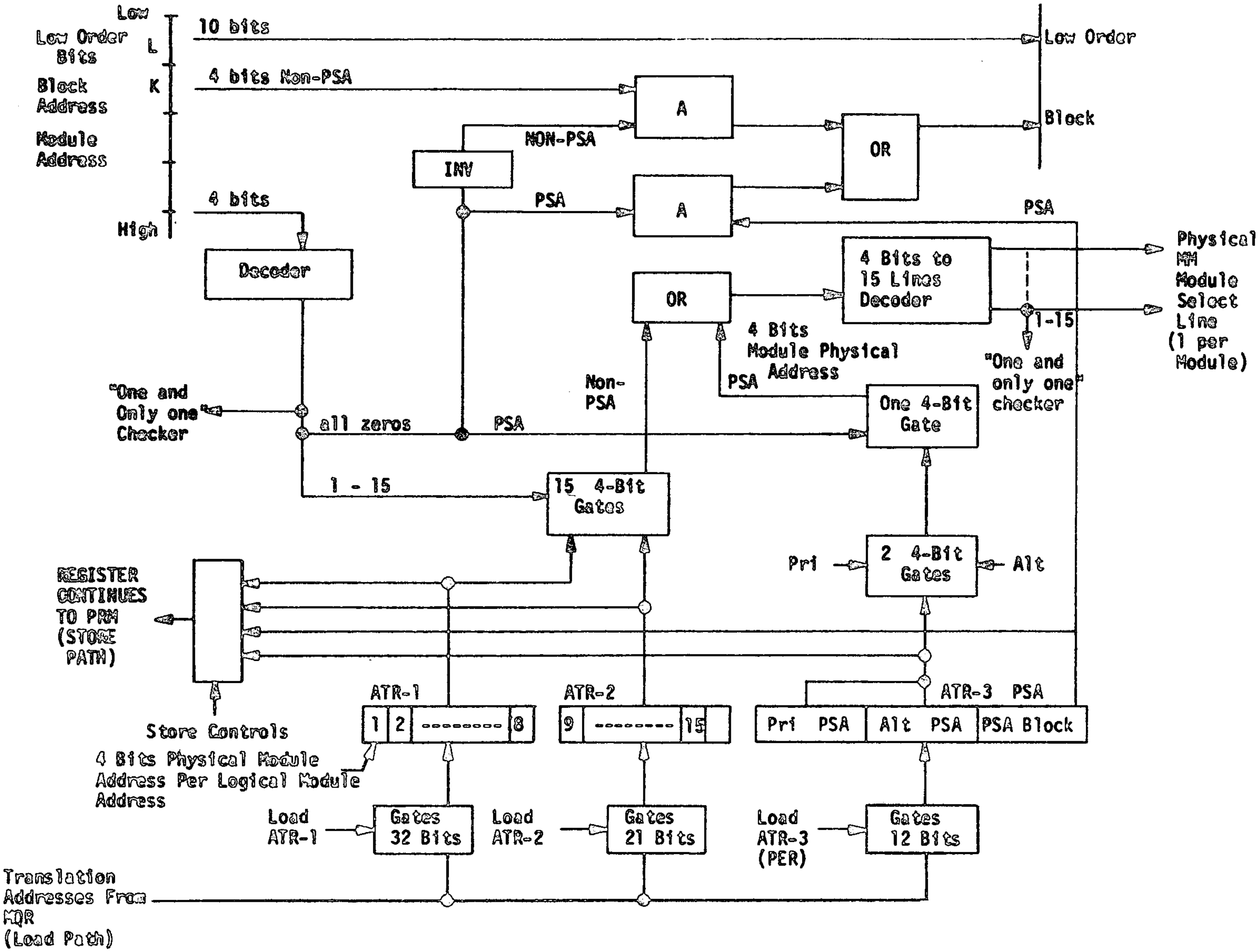


Figure 2-23. ATU Block Diagram

accessible to the requesting CPU. This is accomplished by testing the status of a bit in a status indicating register representing the status of the block in question.

A Delay instruction calls for a programmed delay of n microseconds as specified in the instruction. This instruction is useful following a test and set instruction to avoid repetitive attempts to access a particular storage block when that block is unaccessible.

A Store Identity instruction is used to identify to the executive program which CPU is performing tasks defined by the executive.

Direct Control Lines

A number of direct control lines must be added between each CPU and all other CPU's, IOC's or main memory modules in the system. These lines perform the following functions:

A. Control the loading and storing of data for the configuration control unit registers.

B. Command all other elements to the reconfiguration state under direction of a CPU controlling the reconfiguration process.

C. Select (via address translation units associated with each CPU and IOC) the memory module to be addressed.

D. Provide response signals from main memory module to IOC or CPU indicating that a memory access request has been received and is being executed.

E. Control the loading of storage protection keys in main memory modules.

F. Control power up and power down operations for each element in the system.

G. Receive interrupt signals from other system elements.

Configuration State Monitor

A configuration state monitor is required to assure that the system is not locked up in the reconfiguration state. The configuration monitor tests to see if the reconfiguration state has been cleared up at a certain time after the reconfiguration state was initiated. If not, a signal is sent out to notify the crew or four ground stations via telemetry. A detailed description is given in Section 2.1.3.3.

2.1.4.2 Interrupt Processing

Interrupt processing is required for reasons such as:

A. Service input/output channel upon completion of message to/from data bus to start next message.

B. Resolve problems when parity or memory protect errors are detected by other elements.

C. Protect vital memory data during power failures or transients.

Interrupt processing may be implemented in either of two ways:

A. Single-level, in which all interrupt sources have the same level of priority

B. Multiple-level, in which some kinds of interrupts have higher priorities than others, and thus are processed first.

The single level interrupt concept is illustrated in Figure 2-24. This concept requires the addition of an interrupt signal register, mask control register, interrupt address register, and associated control logic. Operational details are discussed in Section 2.1.3.

The multiple-level interrupt concept is illustrated in Figure 2-25. This concept provides for interrupt of normal operations as well as interrupt of lower priority interrupts in process, and requires replication of scratch-pad memory for each level of interrupt and an SPC select register to select the SPM according to the level of interrupt to be serviced. Mask control is on an interrupt source basis, and lower level interrupts are inhibited until higher-level interrupts are processed . The multiple-level interrupt details are discussed in Section 2.1.3.

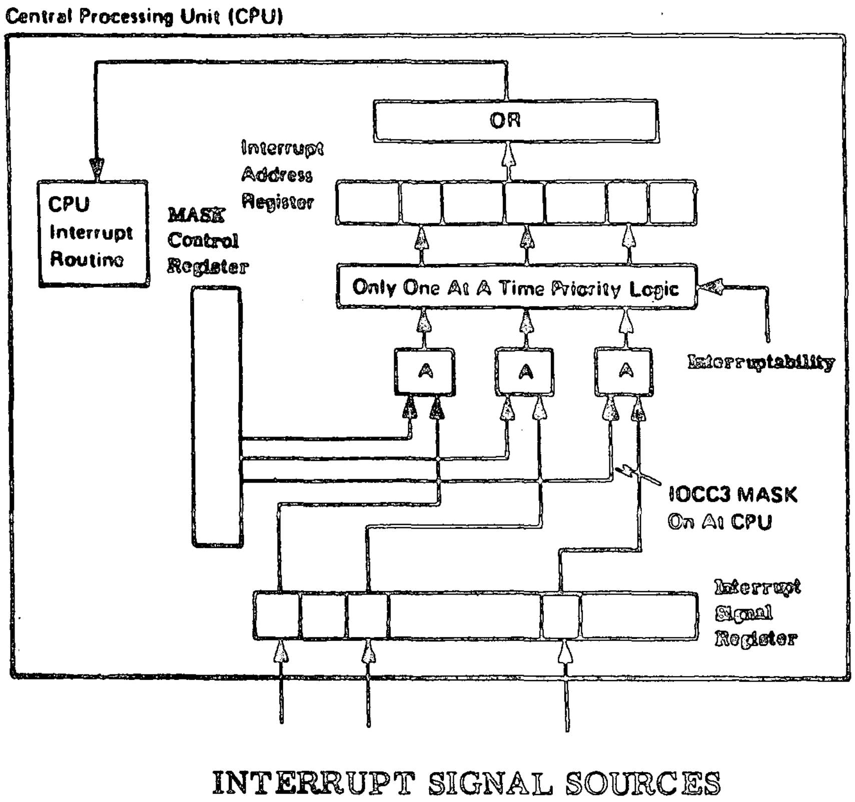

Central Processing Unit (CPU)
OR
Interrupt Address Register
CPU Interrupt Routine
MASK Control Register
Only One At A Time Priority Logic
A
A
A
Interruptability
IOCC3 MASK On At CPU
Interrupt Signal Register
INTERRUPT SIGNAL SOURCES

Figure 2-24. Single Level Interrupt

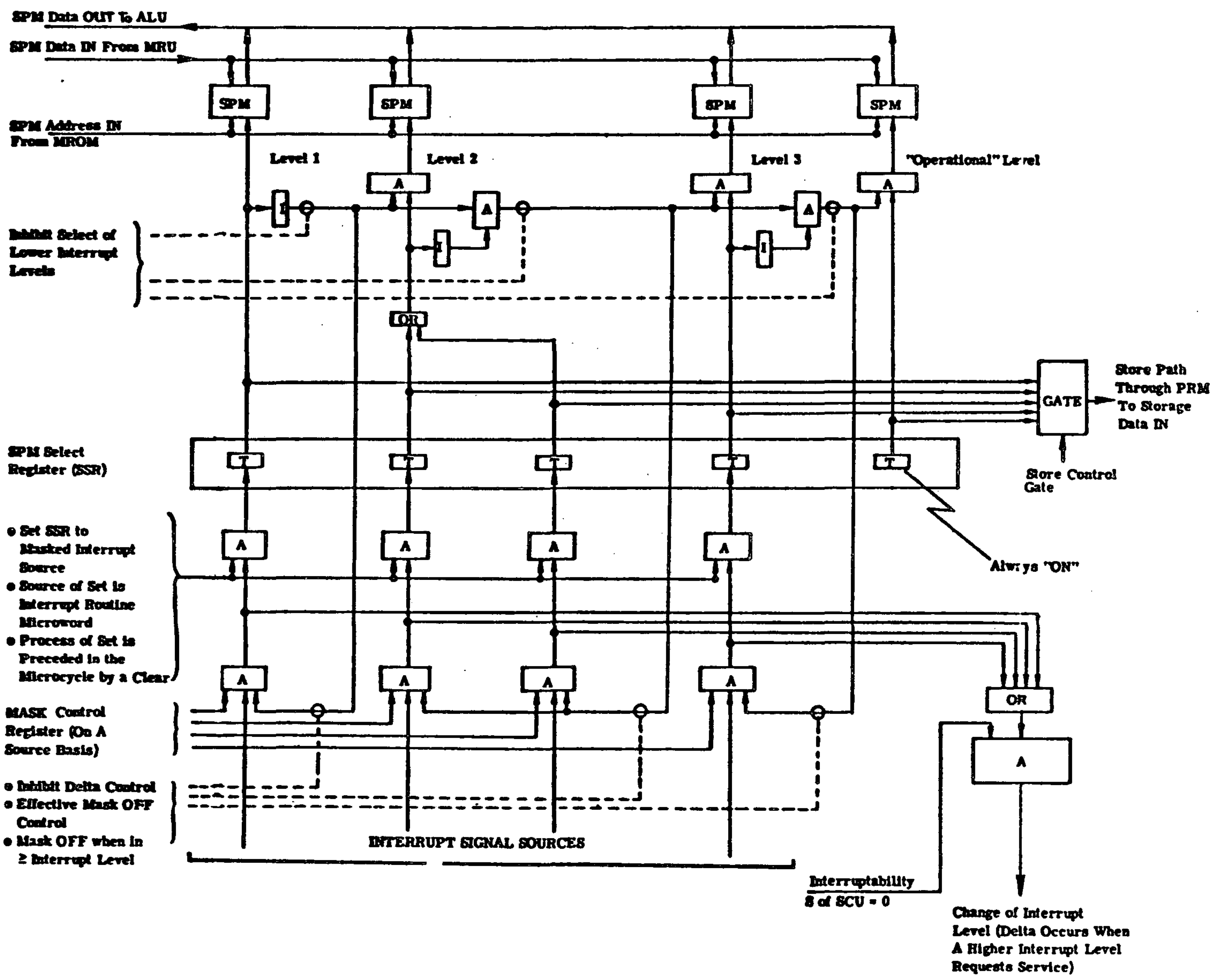

SPM Data OUT To ALU
SPM Data IN From MRU
SPM Address IN From MROM
SPM
Level 1
Level 2
Level 3
"Operational" Level
A
I
OR
Inhibit Select of Lower Interrupt Levels
GATE
Store Path Through PRM To Storage Data IN
Store Control Gate
SPM Select Register (SSR)
T
Always "ON"
• Set SSR to Masked Interrupt Source
• Source of Set is Interrupt Routine Microword
• Process of Set is Preceded in the Microcycle by a Clear
MASK Control Register (On A Source Basis)
• Inhibit Delta Control
• Effective Mask OFF Control
• Mask OFF when in ≥ Interrupt Level
INTERRUPT SIGNAL SOURCES
Interruptability
8 of SCU = 0
Change of Interrupt Level (Delta Occurs When A Higher Interrupt Level Requests Service)

Figure 2-25. Multiple-Level Interrupt

2. 1. 4. 3 Storage Protection

Storage protection is required to prevent unauthorized access to protected blocks of storage due to programming errors or equipment malfunctions. The block size which will be protected has not been determined but would probably be in the range of 64 to 512 words. The storage key is six bits; one for storage protect, one for access protect, and four for the key if protected. Each memory module must have a separate part of storage or non-volatile registers for holding these six-bit storage protect words. The CPU provides its four-bit key to the memory as a four-bit extension of the memory address. The CPU loads the key word area of the memory modules by a special privileged instruction. Thus, one of the memory operating modes is store protection key.

2. 1. 4. 4 Privileged Instructions

Privileged instructions are those which can only be used by the executive program. A modification is required in the CPU microprogram to identify these instructions and inhibit their unauthorized use due to programming errors or equipment malfunctions. An extra bit in the IAROM is proposed.

2. 1. 4. 5 Input/Output Instructions

Special input/output instructions are required to interface with the input/output controllers. These instructions are:

A. Begin Input/Output.

B. Halt Input/Output.

C. Test Input/Output.

The Begin Input/Output instruction is used to initiate a particular input/output sequence. Detailed execution of the sequence is accomplished by the input/output controller operating in conjunction with the main memory for detailed instructions and data transfers.

The Halt Input/Output instruction is used to stop operations on the particular IOC channel addressed.

The Test Input/Output instruction is used to read the input/output controller status register to determine the status of each channel in the IOC.

Further details are contained in Section 2. 1. 2.

2. 1. 4. 6 Partial Word Manipulation

If a significant amount of partial word processing is utilized (byte or half-word) to reduce memory requirements then processing time should be reduced by the inclusion of direct shifting main memory and the CPU.

The shifting could be implemented in conjunction with memory input multiplexing in the CPU to accomplish byte or halfword shifting. A half-word arithmetic right shift currently exists, but a left shift as well as logical right shifts are desirable for word packing and unpacking.

2. 1. 4. 7 Interval Timer

An interval timer should be provided for operation in conjunction with each CPU. The interval timer would be used to accurately schedule future processes without extensive software overhead. The interval timer should be programmable to interrupt the CPU for times between 1 millisecond and 64 seconds after timer start.

2.2 INPUT/OUTPUT SYSTEM DESIGN

2.2.1 Requirements

The Input/Output Controller is required to perform the following functions:

A. Control all data bus information flow.

B. Control the acquisition of data for onboard data processing.

C. Control the transmission of data from the digital computer and main memory to other subsystem elements.

These functions are performed by two units:

A. Computer Interface Unit (CIU).

B. Subsystem Interface Unit/Computer (SIU/C).

All IOC functions are under the supervision of the Central Processor Unit (CPU).

2.2.1.1 Computer Interface Unit

The CIU is required to:

A. Receive, decode, and control the execution of input/output commands initiated by the CPU.

B. Fetch detailed input/output control sequences from main memory and execute these control sequences.

C. Fetch data from main memory for transmission to the data bus via the SIU/C.

D. Serialize commands and data to be transmitted.

E. Deserialize information received from the data bus.

F. Provide for transmitting data and commands simultaneously to as many as three data bus channels.

G. Provide for receiving data simultaneously from as many as three data bus channels.

NOTE: For items F and G the total number of data bus channels is three. Any one channel may transmit or receive on the same frequency, but not simultaneously.

H. Provide for controlling and performing bidirectional communications with auxiliary memory units.

I. Provide buffer storage for data and commands being communicated to and from the data bus.

J. Control the storage of information received from the data bus into main memory.

K. Provide timing and internal synchronization.

L. Signal the CPU upon termination or completion of data communications.

M. Generate parity bits and test for parity for information stored and fetched from memory.

N. Provide for testing of IOC operational status under CPU control.

O. Monitor status of data integrity and signal the CPU upon failure of integrity tests.

2.2.1.2 Subsystem Interface Unit/Computer

The SIU/C consists of three sets of transmitter and receivers and associated modulation, demodulation, circuitry, filtering, and signal conditioning called "modems." Each modem operates on a carrier frequency that is separate from other modems and will use the same carrier frequency for both transmitting and receiving. Each modem will be capable of transmitting and receiving digital information at rates as high as 10 Mbps.

The functional requirements of the modulator-transmitter portion of each modem are:

A. Apply premodulation filtering and signal conditioning to input signals from the computer interface unit.

B. Modulate the modem carrier.

C. Filter the modulated output to confine the signal to its allocated bandwidth.

The functional requirements of the receiver-demodulator portion of each modem are.

A. Filter the modulated carrier input from the data bus according to carrier frequency and bandwidth requirements.

B. Demodulate the modulated carrier input.

C. Condition the digital output signal to specified voltage levels.

2.2.2 Simplex Input/Output Controller Configuration

The SUMC IOC provides the functional capability of interfacing the SUMC CPU with the data bus. The IOC is composed of two major blocks: the Computer Interface Unit (CIU) and the System Interface Unit/Computer (SIU/C). The SIU/C is a set of modems and associated circuitry which interfaces the CIU with the data bus. The CIU provides the controls and timing to interface the CPU with the SIU/Cs. The relationship between computer, memory, IOC, and data-bus elements is indicated in Figure 2-26.

2.2.2.1 Computer Interface Unit

The CIU consists of three units. Program Control Unit (PCU), Input/Output Channel Control (IOCC), and the Bus Channel Control (BCC). The PCU provides the control and timing to and from the CPU/MM and the attached channels. The IOCC—under PCU control—sequences, fetches, and receives data/commands from and to MM and buffers them for the BCC. The BCC serializes/deserializes the data/commands to/from the modem and IOCC.

Program Control Unit (PCU)

The Program Control Unit (PCU) is the control center between the computing and memory element and the channel control elements of the processor I/O function. The major elements of the CPU are the program control, memory input and output multiplexers, status monitor/register, self test control, and timing and control (overhead). The PCU controls requests to and from main memory and associated parity checking and generation. Gating controls for the memory input/output multiplexers are generated by the PCU. The status of the channels and the PCU are maintained in the status monitor/register portion of the PCU.

Input/Output Channel Controller (IOCC)

The IOCC provides the CIU with the capability of sequential data/command transmission or reception without direct program intervention. The IOCC

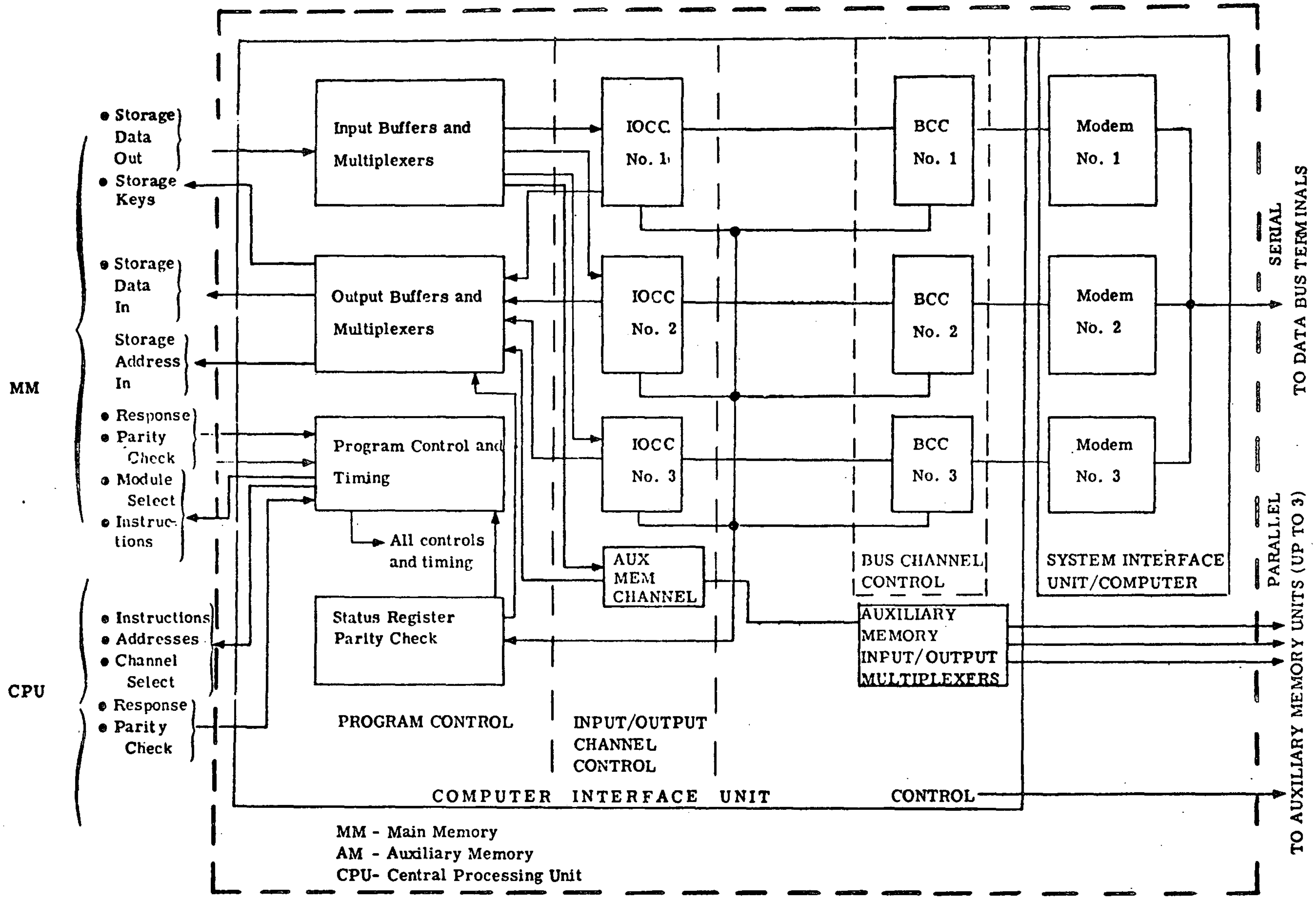


Figure 2-26. Simplex I/O Controller

controls the sequencing of memory address for fetch or store of data between main memory and the data bus or auxiliary memory. The major elements of the IOCC are the Sequence Address Register (SAR), Next Instruction Register (NIR), adder, address and command decoder, input and output buffer registers, and timing and control.

Bus Channel Controller (BCC)

The Bus Channel Controller (BCC) is the interface between the parallel IOCC and the serial DBT functions. The BCC serializes 18-bit parallel words to be sent out over the data bus and converts serial words from the data bus to parallel words for storage in memory. The major elements of the BCC are the input shift register, output shift register and the BCC control.

2.2.2.2 System Interface Unit/Computer (SIU/C)

The SIU/C modems and interfaces with the BCC's in the CIU and the data bus are indicated in Figure 2-27.

The SIU/C consists of three sets of transmitters and receivers and associated modulation, demodulation, circuitry, filtering, and signal conditioning called "modems." Each modem operates on a carrier frequency that is separate from other modems and will use the same carrier frequency for both transmitting and receiving. Each modem will be capable of transmitting and receiving digital information at rates as high as 10 Mbps.

The transmitter unit consists of a crystal-controlled carrier oscillator, a balanced modulator, and premodulation filter and signal conditioning section, and an output filter. The receiver unit consists of an input filter, rf amplifier, envelope detector, baseband buffer amplifier, and a logic threshold detector section.

Automatic gain control (AGC) is shown as an option to be incorporated in the event that the input rf dynamic range exceeds the receiver's capability without gain adjustments.

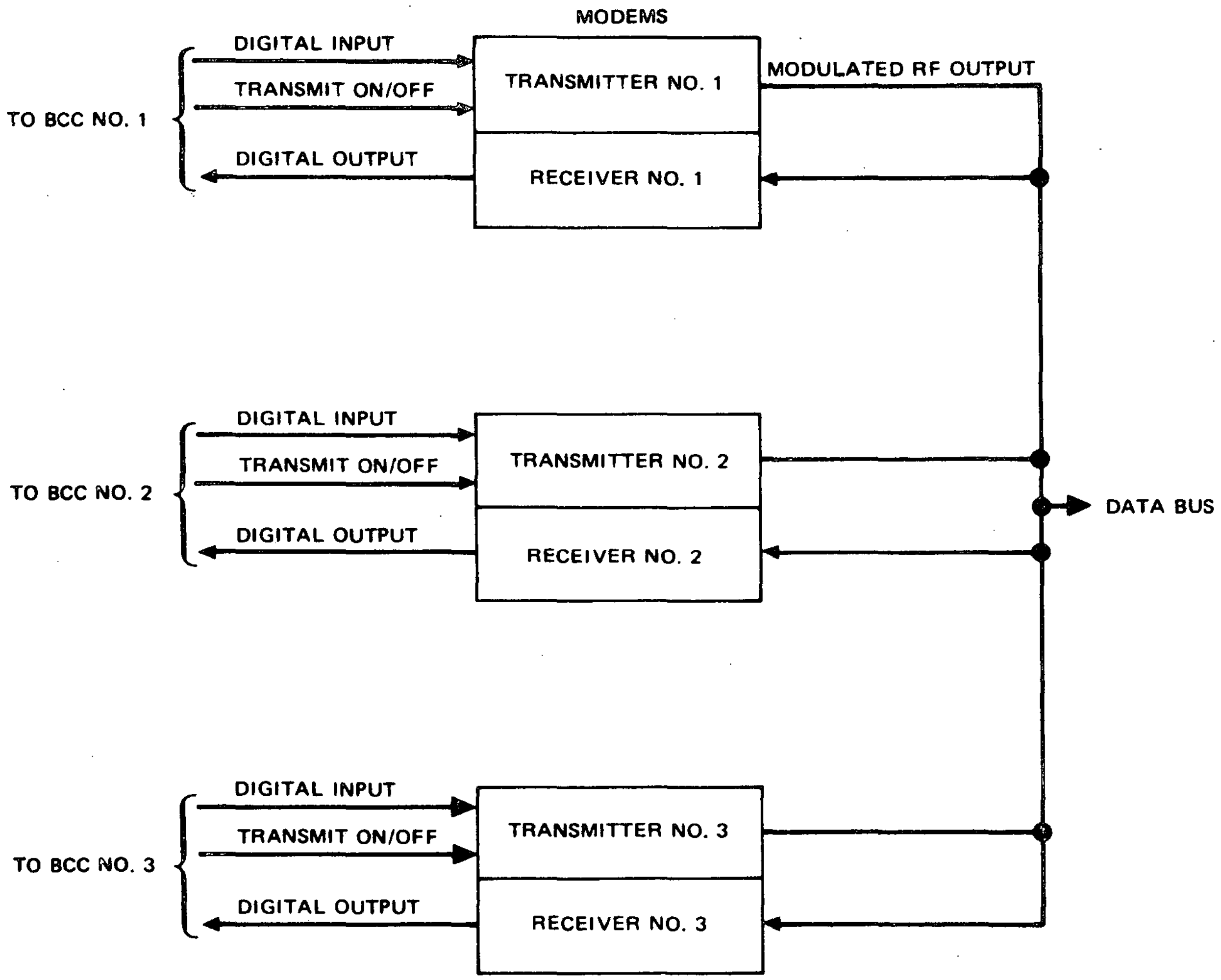


Figure 2-27. Subsystem Interface Unit/Computer (SIU/C)

2.2.3 Multiprocessor I/O Configuration

This section describes the expansion of the simplex IOC described in Section 2.2.2 to become a part of the multiprocessor system described in Section 2.1.3. An understanding of those two sections is required for full understanding of this section.

2.2.3.1 Configuration Control Unit (CCU)

To expand the simplex IOC to a multiprocessor IOC requires only the addition of a CCU. The CCU performs the following functions:

A. Translates a logical address to a physical address.

B. Establishes the units with which the IOC is allowed to communicate.

C. Identifies the state or mode of operation: OFF, OPERATING, OFF-LINE, etc.

D. Halts operation in the presence of a state 3 (reconfiguration) signal from one of the CPU's.

E. Absorbs the storage key logic from the simplex IOC.

F. Provides the starting address of a preferential storage area in memory.

Address Translation

The address translation within the IOC is the same as described in Section 2.1.3.3 for the CPU.

Communication Control

The Communication and State Register (CSR) has 21 bits to indicate enabling of communication with the 15 MMU's four CPU's, and two mass memory units (if required). This is very similar to the communication control described in Section 2.1.3.3.

State Control

Two bits of the CSR are used to indicate the state of operation of the IOC. The states for the IOC are:

State 0	(00)	Off-line/maintenance
State 1	(01)	Powered down/non op
State 2	(02)	Operating

Functioning of the states is as described in Section 2.1.3.3.

CCU Loading

The registers of the CCU are loaded by the CPU with a LCCU/E instruction which requires that the CPU be in state 3 and therefore the IOC will not be operating. As described in Section 2.1.3.3 the data for the CCRs come from main memory on the memory data bus. The signals to control the IOC's ingating to load the registers comes directly from the CPU which is reconfiguring the system. Since only one CPU can be in state 3 at a time, these control signals are tied together in "DOT ORs". A corresponding SCCU/E allows storing of the CCR's.

2.2.3.2 Interrupt

Since interrupts are an I/O function but are mechanized in the CPU, they can reasonably be described in either the CPU or I/O section of the report and the I/O has been selected. Except for the sources of the interrupts, there is no difference in the interrupt structure of a simplex computer and a multiprocessor.

A single-level interrupt system and a multiple-level interrupt system are described in Section 2.2.3.2. Each system is responsive to interrupt signal sources.

The interrupt signal sources include:

A.	Input/Output Channel Controller (IOCC) (one per digital channel)	-	8
B.	CPU's (includes interrupt to self)	-	4
C.	Privileged instruction	-	1
D.	MM's	-	15

Each system includes the ability to mask the interrupt. If masked ON, the interrupt will be processed by the CPU in time and in accord with the established system priorities. If masked OFF, the interrupt will not be seen by the CPU. The mask is under executive control and can be loaded into the mask control register by instruction from PSA. Masking in both systems is on a signal source basis.

The interrupt signal sources request service asynchronously with respect to CPU processing. The mask is an overall gate controlling what the CPU will inspect. Inspection of the sources occurs under CPU logic control during an interruptability period. It is during the interruptability period that the CPU logic samples the interrupt signal sources. The interruptability period for both systems, single-level or multiple-level, is defined as the first microcycle of instruction fetch.

Single-Level Interrupt System

The single-level interrupt system is shown in Figure 2-28. The interrupt system makes extensive use of the PSA concept. The single-level system will be described in two parts. The first part will pertain primarily to the IOCC-associated interrupt logic. The IOCC part of the single-level is also applicable to the multiple-level system.

The interrupt logic shown for the IOCC provides for automatic store into PSA on any IOCC status change when masked ON at the IOCC. The "automatic store on any status delta" is an independent function which does not interfere with CPU processing. The interrupt signal when masked ON at the CPU will in time and in accord with the established priorities actually interrupt CPU processing.

The interrupt system for the IOCC provides for inspection of status independently of interrupt to the CPU. The sensing of a change in IOCC status activates two separate sets of logic. One set is used to signal the CPU and is the normal interrupt to the CPU. If the CPU is masked ON to listen, processing will be interrupted to service the interrupt. The other set of logic causes an automatic store into PSA in accord with masking at the IOCC. The interrupt system servicing the IOCC effectively separates the automatic reporting into MM of status change from that reporting typically associated with interrupt of CPU processing.

The IOCC interrupt system includes the following features:

A. Permits stacking of interrupts till CPU can get to them.

B. Permits identification of interrupt on a per IOCC basis.

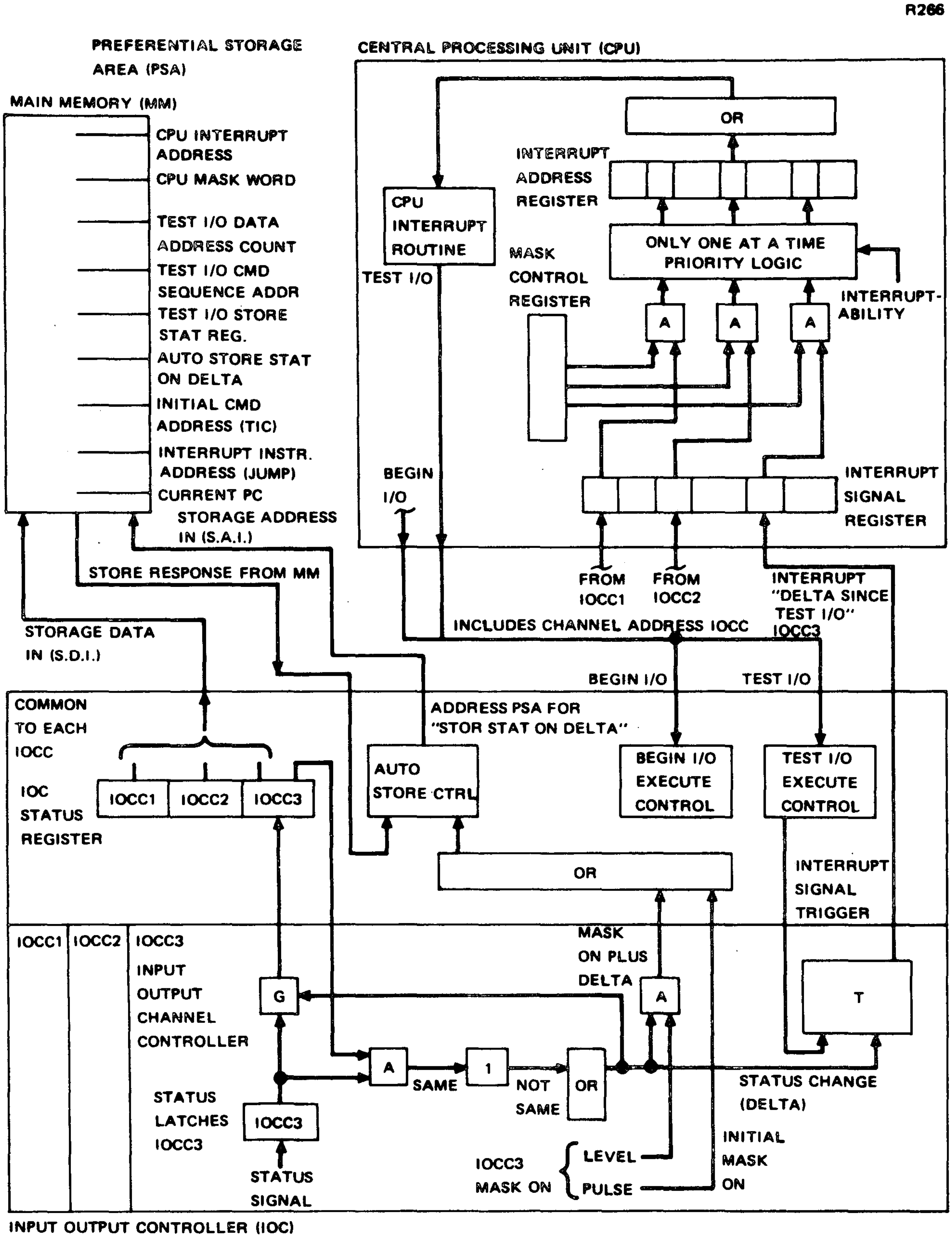


Figure 2–28. Single Level Interrupt System Diagram

C. Permits clearing of interrupt and IOCC status on a per IOCC basis.

D. Permits individual IOCC masking both at the CPU and at the IOC.

E. Permits hold of status of any IOCC till cleared via test I/O.

F. Provides automatic store into preferential storage area (PSA) on any status delta when masked ON at the IOCC independently of CPU processing.

G. Permits inspection of IOCC status without going via I/O instruction to the IOCC.

H. Permits inspection of IOCC status without clearing.

I. Permits any CPU in a multiprocessor system, by reading the PSA, to inspect IOCC's status.

J. Utilizes the PSA extensively to effect interrupt, provide for status reporting, and enhance inter-CPU communications.

K. Permits inspection of one IOCC status through mask ON at another IOCC.

L. Provides the interrupt signal to the CPU on any status delta since last test I.O.

M. Provides for storage into PSA of preprogrammed IOCC control and status contents prior to clearing the IOCC status thus reported.

N. Provides masking of the interrupt to the CPU at the CPU separately from the masking of the auto store to PSA at the IOC.

The IOCC interrupt system includes the following operations, which are descriptive of the system:

A. The IOC status register is set by each of the 3 IOCC's separately. The IOC status register (SR) is divided into 3 10-bit IOCC registers. The entire IOC SR is stored with any store into MM.

B. The store of the IOC SR is in two ways:
   1. Auto store on any status change or delta.
   2. Store on a Test I/O.

C. The auto store on any delta is inhibited during any test I/O. The auto store on any delta is maskable by the IOC on an IOCC basis. The IOCC mask at the IOC is applied or removed in accord with IOC/data bus terminal "A" type commands.

D. The auto store on any delta is to a PSA store status on delta. This PSA location is assigned for each IOC in each of the CPU's PSA's.

E. The use of auto store into PSA allows any CPU to inspect channel status of all three IOCCs with one PSA access.

F. The store on a Test I/O is to a PSA designated Test I/O Store Status. This PSA location is available once in each of the CPU's PSA. This PSA is composed of three contiguous locations. The three contiguous locations are designated:

1. Test I/O Store SR.
2. Test I/O Store IOCC SAR (command sequence address register).
3. Test I/O Store IOCC Data Count.

G. The reason for a separate PSA for Test I/O is to protect Test I/O-retrieved contents from an overwrite by the auto store on delta system.

H. Test I/O on an IOCC basis stores entire IOCC SR, during store inhibits introduction of any status delta from any IOCC, upon store of SR clears SR on an IOCC basis; with clear of IOCC part of SR also clears corresponding status signals stored in status latches in respective IOCC; upon completion of execute of test I/O releases hold on delta input to SR. During test I/O and before release on delta input to SR the interrupt signal trigger in the IOCC is reset.

I. The interrupt signal since last Test I/O based on a delta since the last Test I/O is stacked along with the SR contents until CPU is interrupted and directs the store on an IOC basis, and the clear on an IOCC basis.

J. The status since last Test I/O can be obtained by any of a number of means:

1. Mask ON at the IOC any one IOCC and reference the PSA.
2. Mask ON at the CPU any one IOCC's interrupt and reference the PSA.
3. Use a Test I/O instruction.

The interrupt logic shown for the CPU for single level is brought into play on interrupt and masked ON. The interrupt signal when masked ON at the CPU will in time interrupt CPU processing. When processing is interrupted, the interrupt system automatically will first provide for subsequent continuation of the processing upon completion of interrupt service, and then will proceed to inspect, service, and respond in accord with the specific interrupt signal source.

Upon completion of interrupt service, an instruction "Terminate Interrupt" actually returns the CPU to the processing from which it was interrupted. Prior to the return to the interrupted processing, housekeeping needed to prepare for the next interrupt is essential. Should there be a pending interrupt, the single-level interrupt system will continue to service the interrupts one at a time and in accord with the established priorities.

The interrupt to the CPU is maskable at the CPU on a signal source basis. When masked ON one highest priority masked ON interrupt gets through the priority circuits to call for a CPU branch to the I/O interrupt routine. One action of the interrupt routine in the case of an IOCC interrupt is to issue aTest I/O to the IOCC.

The interrupt address register word defines the source of interrupt. In the case of an IOCC interrupt the interrupt address word from the IAR in the PSA contains the IOCC address being serviced. The interrupt address is stored as a part of the interrupt routine and together with the contents of the PSA for Test I/O provides the status input for analysis, as part of and subsequent to the interrupt routine. Before the next Test I/O, the status information and its interrupt address should be relocated in MM thereby preparing the area for the next interrupt.

During the interrupt routine, no further interrupt is allowed through to the interrupt address register. The CPU mask control register in the CPU is forced to a mask OFF position till ready to mask ON again per contents of PSA CPU mask word.

The interrupt signal register holds the IOCC interrupts pending and can be inspected by CPU Store type instructions either as part of or subsequent to interrupt routine.

The mask at the CPU is obtained from a PSA location and loaded by Load to CPU Self instruction. This mask being in PSA is inspectable and settable by any CPU.

The CPU interrupt routine includes the following operations:

A. PSA for program counter (PC) is cleared for use prior to entering the interrupt routine.

B. CPU masks ON for interrupt.

C. Interrupt signal is received.

D. Interruptable point in the instruction/execution cycle is reached.

E. Transfer to interrupt routine is effected.

F. Microcontrolled part of interrupt routine:

1. First Microaction
   Decrement SPM_{pc} (if in ROM 1 of fetch the MQR to SPM_{pc} could not be inhibited, decrement)
2. Second Microaction
   PSA address for PC to MAR.
   Store SPM_{pc} to PSA.
3. Third Microaction
   PSA address for IAR to MAR.
   Store IAR per MAR.
   Reset mask word masking OFF further interrupts.
   Clear IAR.
4. Fourth Microaction
   PSA address for interrupt jump instruction to MAR.
   Load jump instruction into SPM_{pc}.
   (Upon exit from interrupt routine and transfer to fetch zero will automatically jump to location to complete interrupt processing while masked OFF.)
5. Fifth Microaction
   Go to ROM 1 of fetch zero which will take you to the jump instruction location.

G. Instruction controlled part of interrupt routine:

1. Instruction: jump.
2. Instructions: Do in accord with the specific interrupt per IAR. Include test I/O if IOCC interrupt. Protect and restore preinterrupt SPM contents if disturbed.
3. Instruction: Terminate interrupt.
   Load PC from PSA to SPM_{pc}.
   Load mask from PSA.

H. Return to operational program processing (unless masked ON interrupt is pending).

The interrupt system of Figure 2-28 for single level includes functional registers which make up the Interrupt System Unit (ISU). The functional registers include the Interrupt Address Register (IAR), the Mask Control Register (MCR), and the Interrupt Signal Register (ISR). The load and store paths used for the CCU loading are also used to load and store the ISU as shown on Figure 2-29.

Store of ISR is by instruction. The MCR will be loaded and stored by instruction. The loading of all zeros in MCR, effectively resetting MCR, is under control of the interrupt microroutine. The IAR is stored automatically as a part of the interrupt microroutine. Upon store of the IAR it will be automatically cleared.

As shown on Figure 2-29, the ISR is auto reset following any store. It is up to the program, therefore, to remember those interrupts thereby "wiped out" and not yet serviced. There will be interrupt signals that will not be wiped out following store of ISR, such as interrupts from IOCC. In this latter case, the ISR does not contain a latch for the corresponding IOCC interrupt, but contains only a receiver. Thus only when the IOCC interrupt is reset at its source will it clear. In the case of a CPU to CPU interrupt the interrupt signal is latched and only by storing can the signal be cleared.

Multiple-Level Interrupt System

The multiple-level interrupt system is shown in Figure 2-30. The interrupt system makes extensive use of the PSA. The interrupt system is compatible

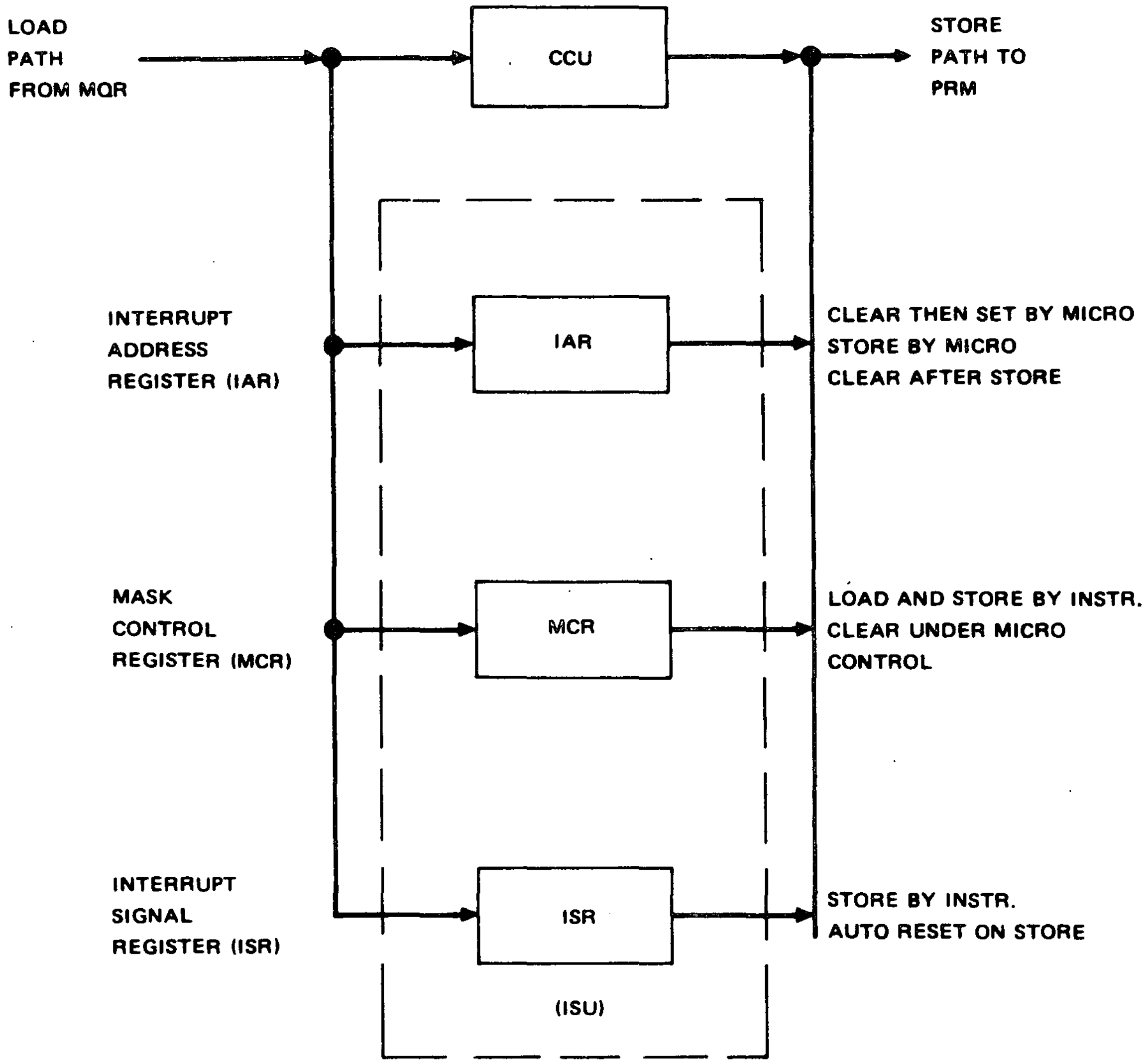


Figure 2–29. Interrupt System Unit Diagram, Single Level

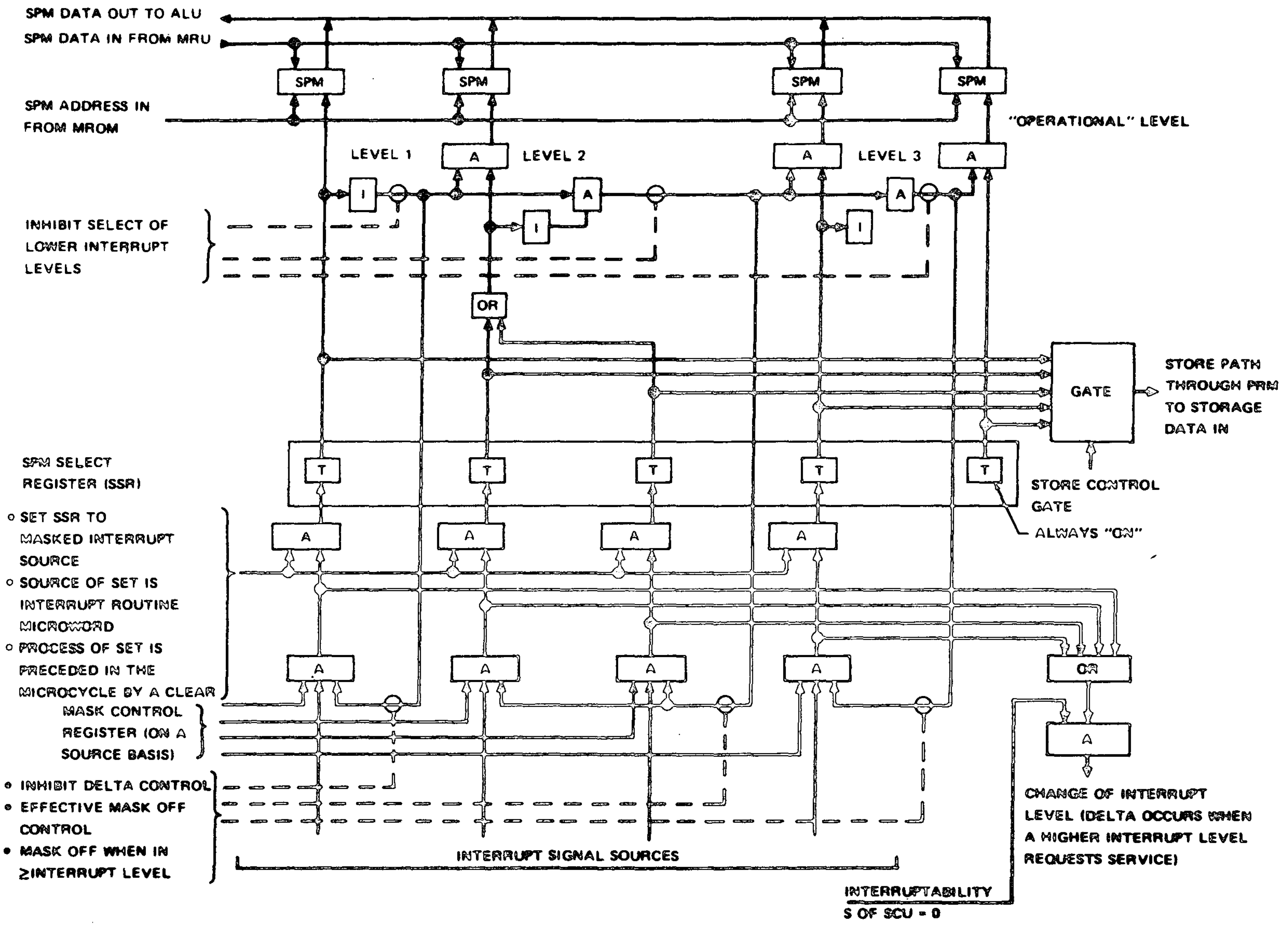

SPM DATA OUT TO ALU
SPM DATA IN FROM MRU
SPM ADDRESS IN FROM MROM
SPM
SPM
SPM
SPM
"OPERATIONAL" LEVEL
LEVEL 1
LEVEL 2
LEVEL 3
A
I
OR
INHIBIT SELECT OF LOWER INTERRUPT LEVELS
GATE
STORE PATH THROUGH PRM TO STORAGE DATA IN
SPM SELECT REGISTER (SSR)
T
STORE CONTROL GATE
ALWAYS "ON"
o SET SSR TO MASKED INTERRUPT SOURCE
o SOURCE OF SET IS INTERRUPT ROUTINE MICROWORD
o PROCESS OF SET IS PRECEDED IN THE MICROCYCLE BY A CLEAR
MASK CONTROL REGISTER (ON A SOURCE BASIS)
• INHIBIT DELTA CONTROL
• EFFECTIVE MASK OFF CONTROL
• MASK OFF WHEN IN ≥INTERRUPT LEVEL
INTERRUPT SIGNAL SOURCES
CHANGE OF INTERRUPT LEVEL (DELTA OCCURS WHEN A HIGHER INTERRUPT LEVEL REQUESTS SERVICE)
INTERRUPTABILITY
S OF SCU = 0

Figure 2–30. Multiple Level Interrupt System

with the auto store on status delta described as a part of the single-level system. The interrupt system is responsive to like interrupt signal sources as per single-level. The single-level system completed the essential services for one interrupt before returning to the interrupted processing or before beginning service for another interrupt. The multiple-level system provides for the interruption of either operational processing or interrupt servicing.

The multiple-level system incorporates replicated SPM's. Replication automatically provides a save upon interrupt. Each interrupt level is provided with its own replicated SPM and is pointed to by the SPM select register.

The processing at one interrupt level continues until a higher level is masked ON and requests service. The change from one interrupt level to another occurs at the interruptability period. The interruptability period is defined when S of SCU is equal to zero and is effectively the ROM 1 microcycle of instruction fetch.

The sensing of a masked ON interrupt during the interruptability period causes a transfer from instruction fetch to the interrupt microrouting. The interrupt microroutine will do as follows:

A. Clear then set SSR.

B. Store SSR into PSA.

The set of the SPM select register (SSR) during the microroutine is preceded in the microcycle by a clear, effectively allowing any pending masked ON interrupt signal sources to be entered into the SSR.

The interrupt microroutine, upon completion, causes transfer to instruction fetch. The ROM 1 microcycle is entered. The PC from SPM is in accord with the SPM Select Address in the SSR. The highest interrupt level is selected.

The servicing of the interrupt level proceeds under an effective mask OFF of any new signal sources of the same or lower interrupt level. Should there occur a higher-level interrupt signal, the interrupt servicing of the lower level will be interrupted. The interruption again takes place during ROM 1 of any instruction fetch. The presence of a masked ON higher-level interrupt signal source again calls in the interrupt microroutine to clear and set the SSR and to store SSR in PSA. In the case when the lower-level interrupt servicing was not completed, then both interrupt level requests will appear in the SSR, but only the higher level SPM will be selected.

During the interrupt servicing after the interrupt signal source has been reset or cleared, an instruction "Terminate Multiple Interrupt" does as follows:

A. Provides PC to SPM_{pc} to prepare for the next interrupt on exit. PC is obtained from MM using normal addressing.

B. Continues microaction. Transfer to interrupt microroutine to clear then set the SSR and to store SSR in PSA. This microaction provides for the automatic transfer on exit to the highest-level interrupt requesting service.

C. Exit from this instruction's microroutine is to the SPM_{pc} of the selected SPM level per SSR.

The full set of SUMC instructions is available to each level of interrupt. The lowest priority level is the operational level and is that to which the system returns when all interrupt signal sources that are masked ON have been serviced.

Each interrupt level may have sublevels. The sublevels correspond to the signal sources. As any sublevel is serviced, the Terminate Multiple Interrupt instruction is used to prepare for the next interrupt. If during the servicing of a sublevel, and before its completion, a signal source from a higher level appears then the processing of the sublevel of the lower level will be interrupted to service the higher level source. Upon return to the level wherein the sublevel servicing was not completed, it is likely that additional sublevels are now signalling for service. In this regard it is up to the program to supervise the processing of the sublevels within a level.

The need to provide for executive program intervention and to provide for load of any SPM under executive control brings about the requirement for the instruction Load SSR. In the case where the SPM changes are to be made as a part of the interrupt processing for that level, the ability to mask OFF does provide for uninterrupted access to a specific SPM. If the masked OFF condition is continued then the processing will continue for each level until complete and until the operational or lowest level is reached. At that time continuation of mask OFF will facilitate further uninterrupted executive access to any SPM.

To provide for access to any SPM, an instruction "Load SSR" is required. Such instruction under executive control would be preceded by mask OFF. The instruction Load SSR will do as follows:

A. Provide SPM select address for SSR from MM per Z of the instruction to the SSR.
B. Provide PC for selected SPM per SSR contents by referencing PSA under microcontrol to load the selected SPM_{pc}.
C. Upon exit from the instruction, control is transfered during instruction fetch ROM 1 to the selected SPM. The PC previously loaded starts the processing.

To provide for quick return to the operational level, an instruction "Clear SSR for Quick Return" can be used. The instruction will do as follows under microaction:

A. Load all zeros into the mask control register effectively masking OFF.
B. Do remaining microcycles as previously defined for the instruction 'Terminate Multiple Interrupt."

The load and store paths used for loading the CCU are also used to Load and store the ISU for multiple level interrupt as shown on Figure 2-31.

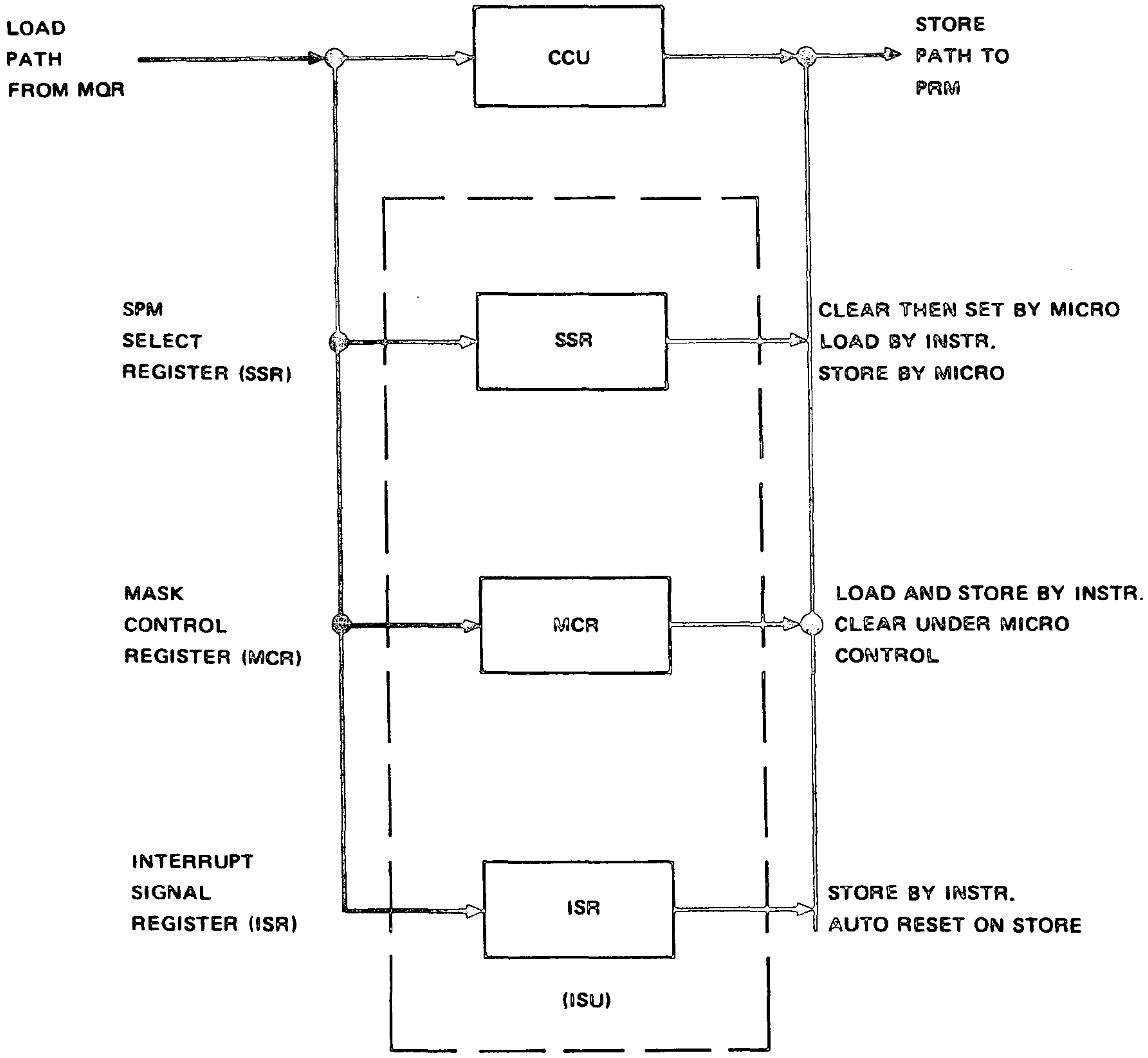


Figure 2–31. Interrupt System Unit Diagram, Multiple Level

2.3 INSTRUCTION SET DEFINITION

The following paragraphs define an instruction set for the Space Ultrareliable Module Computer (SUMC). A major input for the task has been the MSFC Advanced Aerospace Computer, Report No. SP-232-0384, 6 July 1970 (Reference 1); and the Proposed Instruction Set for the SUMC System, SP-232-040501, 4 September 1970 (Reference 2).

The purpose of this section is to define an instruction set to meet the needs of a simplex or multiprocessor configuration of the SUMC and one that has been optimized for the Space Station application. The application requirements have been derived from the baseline study of Modular Space Station.

2.3.1 Scope

The currently defined instruction formats have been followed with minor exceptions as defined in the section discussing proposed extensions to the SUMC design. Emphasis has been placed on the following categories of instructions:

A. Input/output.

B. Mathematical.

C. Bit string manipulation.

D. Multiprocessor-oriented operations.

E. Internal machine-oriented operations.

Processing functions that are expected to be utilized freouently in the Space Station application have been examined and typical algorithms for implementing certain of the functions have been used as examples to demonstrate the power of the recommended instructions. Preliminary computer system concepts have been defined to provide rationale for the selection of other instructions.

2.3.2 Objectives

The major objective is that the instruction set should effectively utilize the internal processor design. To provide selection criteria, effective utilization has been defined as the ability of the instruction to provide for efficient implementation of algorithms utilizing SUMC features while defining independent or dependent application processing functions. That is, functions to accomplish processing for Space Station subsystems or for experiment support.

In general, the SUMC must exhibit many of the characteristics of large ground-based system with multiprogramming capability and yet provide efficient, real-time control associated with spaceborne systems. Some of the common concepts of the ground and space environment required to achieve a reliable and effective system and to arrive at an instruction set definition include:

A. Minimize Application Complexity
   1. Internal architecture for specialized computation.
   2. Executive program interface and capabilities.
   3. Application program and data-structuring techniques.

B. Total System Optimization
   1. Reduce total processing time.
   2. Scratch-pad memory (SPM) utilization.
   3. Reduce program loading or relocation overhead.
   4. Processing monitor, measurement, and development tools.

C. Isolation of Processing Functions
   1. Storage protection.
   2. Privileged instructions.
   3. Interface monitoring.
   4. Resource utilization monitoring.

D. Communication of Processing Functions
   1. Time or event dependency.
   2. Shared data and data handoff.
   3. Multiprocessor considerations.

2.3.3 Summary

The SUMC is designed[1] to handle an eight-bit operation code field, and therefore up to 256 instructions can be immediately decoded and executed in microprogrammed logic. The total number of microinstruction locations

(Read Only Memory) is 1,024, and therefore an average of only four microinstructions per defined instruction if all operation codes are utilized.

The number of instructions specified by this document is only 159. The instruction list (Section 2.3.7.1) and the definitions (Section 2.3.6.2) are grouped by major function and by primary data format or operation.

The totals are summarized in Table 2-4 and, as a group, should provide an adequate and flexible starting point for SUMC and Modular Space Station development activity.

Table 2-4
INSTRUCTION PROFILE

	Fixed Point					
		Floating Point				
Operation / Function	Basic	Single Precision	Double Precision	Double Precision	Special[2]	Totals
SPM content control	22	1	2	0	12	37
Arithmetic (basic)	9	9	9	7	8	42
Bit manipulation	17	1	0	1	2	21
Branching	14	1	1	1	3	20
Special[1]	21	11	1	1	5	39
Totals	83	23	13	10	30	159

Notes: 1. Input/output (3)
Multiprocessor (3)
Arithmetic (12)
Repetitive control (17)
Configuration control (3)

2. Half word manipulation
Branching flexibility
Multiple word SPM operations
Byte manipulation

Sixteen instructions are defined as privileged as indicated by an asterisk in the instruction list (Section 2.3.7.1).

2.3.4 Requirements

The following subparagraphs are discussions of system requirements to meet the objectives outlined in the Introduction and are presented to establish a background for discussion of design concepts in subsequent sections.

2.3.4.1 Minimization of Application Complexity

To reach the full SUMC potential, the processor architecture will have to be configured by appropriate microprogrammed instructions to permit the applications designer or programmer to easily express elements of frequently used programming algorithms. An element will consist of individual or sequences of internal operations controlled by microinstructions. The basic internal operations establish the efficiency of element execution and the level of complexity required to define and utilize elements.

2.3.4.2 Total System Optimization

The optimization of the total system requires that hardware and software features be combined to insure that all operations are performed in the most efficient manner. Therefore, it is a function of (1) the time required to accomplish a basic arithmetic operation, (2) the composition of the basic element groups which comprise the instruction set, and (3) the software interface between the hardware and application programs.

Computation

The basic arithmetic operations, such as multiply, divide, and square root, have not been examined in detail since their processing times compare favorably with those achieved in other processor designs with comparable internal speeds. For instance, without considering the time required to fetch an instruction describing an operation and its parameters or to retrieve or store data words beyond scratch-pad memory, typical times are as follows (32-bit data word):

A. Fixed point add, 1/3 μsec (no data overflow).

B. Fixed point multiply, 3-1/3 μsec (64-bit product).

C. Logical or arithmetic right shift, 1-1/3 to 7 μsec.

D. Floating point add, 1 to 2-1/3 μsec (normalized).

E. Floating point multiply, 4-1/3 μsec (normalized).

Fetching an instruction that selects one of the above operations and acquiring a single data word requires an additional 1-2/3 microseconds. Therefore, the greatest potential for optimizing the instruction set for the SUMC and Space Station application is in the areas of instruction sequencing, internal operation sequencing, acquiring or storing data, and formating data. Extensive data formating will be required to provide compatibility with the basic internal operations of the SUMC and the external environment.

Executive and Application Programs

The executive must provide a reliable and efficient interface between application programs and the SUMC hardware. The following functions are typically provided by executive and/or maintenance programs:

A. Interrupt handling and input/output control.

B. System (SUMC) monitoring, testing, and reconfiguration

C. Application program scheduling

D. Clock and internal timer services

E. Internal communication control

Each of the above involves task switching between the executive and other programs. The instruction set should provide efficient tools for accomplishing each of the functions.

2.3.4.3 Isolation of Processing Functions

In the Space Station environment, where the total processing definition and software implementation will evolve over a 10-year period, it is very important to control the abnormal impact that one software component can have on another. Even if changes are thoroughly evaluated on a ground model of the environment, the application is too complex to expect that problems will not be encountered if some positive action is not taken to control program module interaction. The same holds true for ground based systems and has been successful to the degree that critical resources and intercomponent communication can be controlled.

Critical resources include memory space, processor time, and I/O channel time. A detail discussion is outside the scope of instruction definition. But facilities must be included in the SUMC architecture to accomplish the isolation either directly or in conjunction with the executive program. The concepts and features required will be described for the purpose of defining an instruction set and SUMC extension or possible modifications.

Storage Protection

Storage protection must be provided for either a simplex or multiprocessor configuration. For simplex systems, a bilevel data store lock can be utilized but should be memory block (multiple words) oriented to reduce program task switching overhead. Unless all blocks containing instructions are read-only, then the same bilevel lock techniques must be utilized.

Keys associated with blocks or any address-limiting technique will minimize the lock maintenance overhead. A lock bit per word can present a problem in systems with high task switch rates and a bilevel control will not satisfy the multiprocessor requirement. Key or address limiting mechanisms can be placed in the processors, memory units, or any intermediate unit. A discussion of the latter two approaches as they pertain to the SUMC and store/branch protection is included with base register concepts (Section 2. 3. 5. 2).

Instruction Execution Protection

Certain instructions that can abnormally affect the status of units that make up a computing system or bypass normal program isolation features must be restricted. These are commonly referred to as privileged instructions and by defining internal processor states and controls, only the executive program or portions of the executive can perform the operations.

Several approaches can be taken; from not permitting program self-modification and editing each program before loading; to internal checks to determine if the requesting program is an acceptable user of the instruction. Self-modification or any modification can be controlled by a storage protection scheme. Less precaution could permit an error in any program or hardware failure to inadvertently request and have executed an instruction that could adversely affect the system. (The system should be designed so that multiple hardware failures or multiple software errors are required before functions are erroneously performed.)

An implementation approach for controlling the use of privileged instructions with the SUMC is included with base register concepts. The approach is similar to the store/branch protection logic (Section 2.3.5.2) and assumes the use of a separate set of executive-controlled base registers.

2.3.4.4 External Communication of Processing Functions

The executive program has the prime responsibility for providing the interface to any data management system resource whose uncontrolled use could seriously impair mission success. Examples of such resources are the data bus channels, the timers and clocks, memory space, processor time, and the access to private data sets of individual processing programs.

System communication requires special attention in a multiprocessor configuration and in particular with respect to the constraints placed on subsequent operations by an early scratch-pad memory utilization decision. Shared data controls can be implemented for areas of main memory with a Test and Set (Reference 3) instruction, but means are not available to share scratch-pad memory data at any time without one processor interrupting the other. Shared data and data handoff is not difficult to implement on a simplex configuration but with multiprocessing, requires special instructions when main memory is involved.

Scratch-pad memory (SPM) local to one CPU cannot be communicated to a second with a simple handoff.

To prevent delays, the CPU either has to schedule tasks that normally utilize independent data or an efficient means must be defined to interrupt each other. SPM-to-SPM communication is relatively slow, especially if it involves switching two executing tasks from one CPU to the other. Each CPU should provide the following services to a task without unnecessary interrupting of the other CPU's:

A. Real-time or interval time request.
B. Unique identification to the Executive.
C. Message or data queueing for other tasks.
D. Shared data protection/requests.

2.3.5 Concepts

The following paragraphs define some general concepts that were utilized to determine and shape the instructions listed in this document. Most of the approaches discussed are mutually exclusive, but the section of format definition (Section 2.3.6.1) will discuss the impact of combining certain of the concepts.

2.3.5.1 Memory Addressing

The SUMC design permits manipulation of memory address values with the same magnitude as any other fixed point number. Assuming 32-bit words, the addressable memory (4,294,967,296 words) is much larger than the Modular Space Station baseline which requires less than 18 bits for word addressing (262,144). Even though larger than currently required, 18 to 24 was chosen as an adequate design range and is used as a basis to discuss implementation approaches that can take advantage of the remaining 8 to 14 bits in a 32-bit configuration. Some implementation alternatives are discussed in subsequent paragraphs on address translation and storage protection, but for the remainder of this section, logical data addressing will be assumed.

Many general purpose computers organize memory address to permit addressing below the standard processing word size even though memory access is always the standard word size or multiples of the standard word. Because of the relative difference between internal processing speeds and the time required for memory access, algorithms utilizing dispersed partial words should be avoided. Operations that require access to successive partial words have been given special attention during instruction definition.

Partial Word

One advantage of smaller word sizes is to reduce the total storage requirement for a particular set of data items. The SUMC design permits taking advantage of the smaller words as long as the intraword position during program preparation and the displacement field defined as if full-word references were to be made. Two areas where this would not necessarily hold true are certain experiment data and input or output data to the subsystems or experiment control consoles.

Where a dynamic determination must be made, it is usually associated with an information scan operation or an information packing operation. Both scan (memory fetch) and pack operations (memory store) involve multiple partial word operations and have been defined as internally cyclic microprogrammed instructions where practical. Two common cases of partial word operations are involved with alphanumeric and experiment data.

Alphanumeric packing is only the infrequent transformation of a free format crew input message to a series of word-adjusted input statement segments. The reverse operation is frequent and should be supported with special instructions to provide a flexible capability for generating display formats. For input processing, decimal digit packing should occur during scan or edit operations to reduce program complexity and to provide an intermediate format for efficient microprogram implementation of conversion instructions. Each decimal digit is checked for validity during pack operation, and the reverse or conversion to decimal always generates valid digits but should not be in the packed format, again reducing processing overhead.

Partial word arithmetic operations can be used primarily for experiment data but to a limited extent for on-board checkout and other subsystems-related operations. A further discussion is given under Instruction Set Definition.

Structure

For the remainder of this section, only references to full words (32 bits) will be considered, thus assuming that special instructions will normally be available to satisfy other requirements. Furthermore, when data must be accessed outside the main memory contents, this access will be accomplished through an input/output interface.

The input/output logic will have to be defined to access main memory to acquire data that describe the external placement of data, and therefore the physical address constraints are not of prime concern to the internal SUMC design. What is of concern is the address structure within main memory that relates to program instruction sequencing and to application-program-defined data structures.

Data Structures—To permit systematic buildup of large software systems, it has become a standard practice to provide data descriptions independent of the processing description. During final program preparation steps, these two descriptions are used to determine the actual instruction and instruction address makeup. In the case of the SUMC, the displacement field value, the particular operation code for partial word manipulation, and a base register must be selected.

The Fortran common statements and symbolic equivalent tables processed by assemblers are examples of programming (source) language definitions of data structures. Addresses can be evolved in the successive steps of basic module assembly, linkage editing of sets of modules, the final memory loading process just prior to execution, and during the execution sequence as the actual base register content is loaded. At this point, the address may only be logical and require a final hardware translation step. A subsequent section describes a base register concept that can reduce the complexity of data and base register loading steps, and at the same time permit reconfiguration and relocation logic to be simplified should address translation hardware not be available.

Program Structures—The same address determination steps that were described for data structures are required to establish final memory location of instruction sequences. The complexity appears to increase when controls and features are added to permit the utilization of programming concepts such as reenterable, serially-reusable, recursive, and overlay structures. The instruction set and executive program design must permit the utilization of concepts that have proven successful and are found to be applicable to the Space Station environment.

To achieve one of the earlier-stated goals of minimizing complexity, the addressing technique for program and data structures should be similar, and when practical the set of possible base registers values should be established no later than program load time. Individual program modules need not be permitted to dynamically generate a base register value. The Executive program could maintain a static structure with respect to individual program logic, and, in turn, the individual programs could manipulate base register contents based on a reference structure maintained by the Executive. The instruction set defined does not necessarily impose the rigid restrictions described in this paragraph. Instruction affected by the existence or lack of such restrictions will be identified.

2.3.5.2 Memory Interface

Each program task will normally be given access to three sets of reference points in the address structure under Executive control. A set includes data and instruction (or program module) references in the form of vector lists. Three sets permit immediate reference to elements in the system or top level structure, to the parent or invoking task where appropriate, and to elements currently "owned" by a task.

All base registers can be loaded by special instruction and special formating accomplished by the corresponding microprogram. The Executive program establishes all addressing limits or keys and operating modes and provides the information for the special instructions to preset actual base register values.

Storage Protection

Since the Executive is responsible for the assignment of physical memory address, a method must also be provided for the Executive to dynamically control the protection mechanism. Dynamic control is required to the extent that programs can be relocated or that program tasks may be temporarily assigned store privileges within specific areas of memory.

If program tasks must acquire data area references from an address structure controlled by the Executive and the task itself was created by an invoking task from a similar address structure, then the storage protection mechanism may be referenced to either structure or both. In either case, the mechanism may be a key or any address limiting technique but should not degrade the SUMC performance during normal processing.

The same interface checking should be made between the Input/Output Control (IOC) unit and main memory, and therefore the selected approach must provide for passing the key or address limit to IOC unit. Two basic alternatives are presented in the following paragraphs.

Key Control—If key control is implemented, two instructions in addition to those listed under Instruction Set Definition (Section 2.3.6.2) may have to be implemented. The first is required to precondition the mechanism that compares the key associated with each memory reference. The second instruction is not an absolute requirement if during Executive mode processing, any processor in a multiprocessor system may reset the original key without knowledge of the original key. But its presence will simplify checkout and Executive logic, especially if system partitioning is implemented.

Key control can be a hardware feature in the memory units, even though any point between software control and memory can detect software discrepancies. Checking at the lowest point may also detect discrepancies caused by hardware features. The protection logic checks each memory access, either data or instruction references. If the keys are checked in memory modules, more address lines are required to carry an associated key and usually more hardware is involved because of the number of memory units.

Checking access to memory in the CPU will require the same number of instructions to control the logic. Even in a multiprocessor, each CPU is only executing one program task at a time, and therefore a given set of keys should hold until a task switch occurs. Unless there are an insufficient number of unique keys to support the active tasks, no special problems are expected.

Address Limit Control—A dynamic controlled storage protection mechanism can be implemented in the SUMC processing unit and, if duplicated in other units accessing main memory, will provide the storage protection required for the Modular Space Station. The technique requires that any effective address generated be verified to be within a lower and upper boundary. The lower boundary can be guarenteed if the Executive establishes all base register references and negative indexing is not permitted beyond an instruction displacement value.

Not permitting negative indexing is usually a severe restriction but not serious with the SUMC because of the significant size of the displacement field of each instruction. Negative indexing in excess of the displacement value is relatively simple to detect and upper boundary detection can be implemented with limited changes to the SUMC.

Upper boundary detection can be accomplished by including with each base register value the complement of the highest block of memory to which access is permitted with the associated base address. A block is defined as in the protection key technique, as a set of contiguous addresses participating in a nondivisible protected set. The hardware change requires that the bits corresponding to the block address be duplicated on the base register addition cycle. An eight-bit block address should be chosen because of SUMC four-bit planes. A duplication of block address must occur but may occur at any time after the partial sum ((X) + I) is created. The duplicated field is added to the complement of the upper boundary block to provide a means to recognize effective addresses that exceed the executive defined boundary.

Store/Branch Control—Somewhat independent of controlling the use of addresses is the need to selectively permit the use of certain addresses, depending on the particular operation. The logic may be integrated with the key control approach, and therefore no new instructions are required. The following paragraphs discuss an approach should key control not be utilized.

Independent logic must be provided if the address limit control approach is taken. One approach is for the Executive program to maintain a flag associated with each entry in the base address reference structure. As each base register (1/2 SPM) is loaded, the flag is included as bit 31 and verified with special control logic added to the SUMC.

A signal must be associated with a memory store or branch type instruction and by special control logic, used to recognize a fetch only flag when requested by a particular microinstruction format. A flag must be included with each base register for flexibility. The flag is transferred to the program counter any time a branch operation selects a different base register or any time an effective address is generated for data references.

Address Translation

A requirement for instruction and data to be easily relocatable permits the data management system to be designed to more easily accommodate mission requirement changes, to quickly adjust to various levels of system degradation, and to minimize preplanning required to define dynamic program loading sequences. Relocation requires the addresses of a program to be adjusted to correspond to the physical location assigned by the Executive. Address adjustments may be made by a programmed function within the Executive or be translated by hardware features. The next two paragraphs discuss the two approaches.

Hardware Relocation—The effective address [displacement + (specified index register) + (specified base register)] generated by the SUMC can represent a physical or logical main memory address. The latter permits simplified reconfiguration of memory modules under Executive control without imposing base register restrictions on applications programs or without complex address modification by the Executive during program loading or any reconfiguration process.

If a hardware approach is taken, two instruction types included in those listed under Instruction Set Definition (Section 2. 3. 6) will have to be implemented. As with the storage protection key approach, the first instruction is required to define the logical to physical address relationship. Complete assignment flexibility is provided by a Set Address Translator (Reference 3) instruction which defines the physical equivalent of each possible logical memory unit address. The second instruction retrieves the current value again for Executive simplification and to provide greater multiprocessor flexibility and system partitioning. A similar instruction has been defined for the SUMC in Section 2.3.6.2 (Set Configuration Control Unit).

Software Relocation—Two basic approaches are available for software controlled relocation where hardware address translation features are not available. The first is a totally dynamic approach, like the hardware approach that permits programs to be physically relocated to overcome conditions of fragmentation of a failed memory unit, somewhat transparent to the individual programs affected.

Like the hardware approach, the relocation is only transparent if only recoverable data were lost with a failed unit. Otherwise, a form of checkpoint restart is required if the affected tasks are to continue. Automatic address translation removes the requirement to modify all addresses in other modules that reference the failed unit.

The totally dynamic approach requires that the Executive establish all base references or that base register values be determined in a completely controllable fashion. Application programs can be permitted to adjust base registers but only between limits controlled by the Executive. An alternative is to translate the base register values any time an application performs a base register load operation. A general register to base register load normally takes 2 μsec and the additional lookup would take at least 2 or 3 additional μsec or somewhat less with hardware modifications.

The second approach for software control is to require through conventions that programs be in a relocatable state when in auxiliary storage or in main memory but inactive. The Executive would then recognize states when programs or associated data could be relocated without affecting the program logic. Other states will probably be defined to recognize states which require special restart entry when data is lost through memory failure. These special states need not require special instructions and can be modified with other main memory or scratch-pad memory store instructions.

The second approach may be desirable with address translation schemes in hardware. Without a virtual memory concept, the most that hardware can accomplish is to insure contiguous address even in a partially failed state.

Hardware/Software Relocation—The current design of the SUMC does not permit an efficient implementation of a microprogrammed equivalent of the hardware translation approach. A limited address trap and translate technique could be added that takes advantage of the low probability of having two memory units fail within the period of time required for physical replacement. SUMC changes would have to be made to prevent degradation during normal processing and a totally dynamic relocation solution would not be provided.

One approach requires that the memory unit identification portion of the effective address be routed to the floating point multiplexer for comparison with an execution-controlled value. The value would represent a null or failed unit address and would be brought into the exponent register on an earlier microinstruction and retained until the microinstruction which determines the effective address forces the comparison. If a match occurs, the exponent register value would have to be routed to the equivalent portion of the memory address multiplexer which requires an SUMC modification.

This approach requires a standby module or a module that can be easily taken over under Executive control, preconditioned with backup information, and used as a substitute until the failed unit can be replaced. Contiguous addresses are provided by this approach with a single failed unit without additional address translation hardware. Once notified of the replacement, the Executive can reload the memory and replace the null value.

2.3.5.3 Execution Control States

At least two operating states are required during application processing to control the use of privileged instructions. One state indicates that certain instructions are not permitted to insure system integrity. The states are changed by two special instructions that also modify the execution sequence similar to a closed subroutine linkage technique. Hardware modifications are required.

Primary Control States

One approach requires that the Executive execute in the privileged state and control the flag in scratch-pad memory (SPM) that represents the control state. The flag can be set with a nonprivileged instruction but control would immediately return to the Executive.

To prevent degradation of the SUMC, the flag can be allocated in the same SPM word that contained the next instruction to be executed. To recognize the abnormal condition, special control logic will have to be added that recognizes the most recent flag setting, the issuance of a microinstruction requesting the check, and the concurrent attempt to execute with an operation code defined as a privileged instruction.

The SUMC can be expanded to define a signal with each instruction that must be protected from certain programs. The expansion should include the detection of three signals on the same microprogram execution. The other two are signals generated from a Microprogram Read-Only Memory (MROM) bit selected during the program counter updating and a flag bit stored with the program counter to designate the nonprivileged mode. The process is identical to the previously described store/branch control and as the program counter (1/2 SPM) is fetched for updating, the flag (bit 31) is used as a control signal state.

The Executive address defined to accept transfers from the nonprivileged resulting from the state change must be retained in SPM. As with the state flag, the address must be protected from nonprivileged instruction sequences.

Secondary Control States

Within each primary state, there are several conditions that should be recognized and controlled independent of the current execution sequence.

These can be considered secondary states to the primary privileged or nonprivileged states and a nonprivileged program cannot directly affect the states of privileged programs.

The major difference between primary and secondary is that no special hardware is required for the definition of the latter. Microinstructions can be defined such that a prescribed action can be taken based on the status of one or more flags in scratch-pad memory (SPM). For instance, a set of privileged flags could be defined to permit the Executive program to control the servicing of interrupts.

Another example is special handling of conditions recognized during microprograms such as overflow, underflow, and attempts to take the square root of a negative number. A program can be permitted to define a special address to process exceptional conditions and not be required to repeatedly issue a conditional branch instruction nor have all arithmetic instructions defined as skip type. In either case, processing time is reduced.

To implement the two sets of secondary states, an area would have to be defined in SPM for both privileged and nonprivileged programs. Each area would contain an address or address decrement to be used in the event a microprogrammed branch had to be performed, an inhibit flag to prevent the branch even if the condition does occur, and a flag position to set which represents the condition for a subsequent test in the event that branching was inhibited. At least two instructions—a load and a test type—are required for both privileged and nonprivileged programs.

2.3.5.4 Scratch-Pad Memory Utilization

SPM is made up of 64 words of 32 bits (Reference 1) and provides the major resource for increasing SUMC performance. The current instruction formats indicate a maximum of four bits assigned to each of the three possible registers. Assuming no overlapping assignments of fixed point, floating point, index, and base registers, all of SPM is assigned; but with complete overlap, there would be 48 words for other use.

This section will discuss the concepts that will establish the major guidelines for defining the number of registers, the use to be made of the remainder of SPM, and the types of instructions required to manipulate SPM.

Register Assignment

A large number of instructions can be restricted to a particular type register (one SPM location), but the basic fixed point arithmetic instructions are required on all types except possibly the floating point registers. The total number of instructions can be reduced if all registers are considered as accumulator type and other types are assigned as the same locations or subsets.

Accumulators—A minimum of eight accumulators is desirable if index registers are defined as subsets of the accumulators and floating point is not included. A separate set of accumulators or additional accumulators are desirable if floating point is implemented. For better utilization of high-speed storage used for register storage, the latter or addititional accumulators are more desirable since many programs do not require floating point instructions, and poor utilization of SPM will result. When floating point is used, single precision is used most frequently and it may be possible to take advantage of the other floating point register of the register pair for other processing and storage to minimize main storage accesses.

Assuming special purpose matrix and vector instructions, the number of full-word accumulators should be from 8 to 16, depending on the availability of floating point and the assignment of register types. Load and store type instructions can be minimized if all register assignments are from the same set of directly addressable accumulators. Guidance, navigation, and control (GNC) programs will occasionally require double precision arithmetic operations with either fixed point or floating point operations, and therefore efficient loading and storing of adjacent data words are desirable.

Base Registers—An examination of current and previous Space Station study efforts indicates that if the displacement field is sufficiently large (i.e., ≥ 4K words), then three base registers will most frequently be used. The first is required to maintain the current instruction reference for local branching and to prevent complex address modification during loading. The second is likewise important for loading or the inclusion of dynamic reallocation algorithms and is used to reference local variables or constants. If data or constants that are common to a set of programs such as GNC are stored contiguously, then only one additional base register will normally be required.

Data or constants that are common to more than one set of programs could be accessed using a fourth base register; therefore, it appears that four base registers will suffice without excessive loading of base registers. The displacements will be of sufficient size (12 to 14 bits) to permit the same base register setting to hold throughout the execution of most programs. The concepts discussed in the preceding sections identified a need to define base registers as a separate set of registers or SPM locations to permit controlling their content.

Index Registers—Like base registers, the number of index registers that are required to prevent excessive loading and reloading depends on the number of arrays that are concurrently accessed with independent index variables. Frequently these are no more than two input and one output array, each of which can have an independent index variable. A minimum of three registers is required, and if additional bits are available in the instruction format, extra index registers will be of more value than extra base registers, assuming a sufficiently large displacement field.

In general, the number of index registers should be equivalent to the number of accumulators, since almost any data manipulation can result in an index value for a subsequent operation. Therefore, for programming simplicity and for reducing processing time for requesting data to be transferred from one register to another, the accumulators and index register should be the same set. Unless the current fetch logic (Reference 1) is modified, a transfer requires as much time as a normal fixed point add (2 μsec).

Dual index capability can be implemented on the SUMC without hardware modification in addition to the base register operation for such requirements as matrix inversion. But the more attractive approach appears to be the implementation of special three-dimensional matrix and vector operations as microprogrammed instructions (see Section 2.3.6.2). The instructions will be frequently required in the GNC equations.

Displacement—The common data required by GNC program modules and to interface with other subsystem functions requires a 13-bit displacement assuming half-word addressing. This can be reduced to 12 bits if full-word addressing is utilized and if instructions requiring addressing below the word level have a special operation code containing the additional address bits. The latter approach could increase programming complexity for certain operations, since an addressing scheme should be utilized that addresses the lowest data word size for arithmetic operations.

As discussed earlier (Section 2.3.5.1), special instructions will be defined for common partial word (32 bits) operation. But if the SUMC address concept were changed to address half words (16 bits) or bytes (8 bits), multiple decrement/branch test instructions would have to be defined to keep program loop control at the same efficiency level.

There appears to be no emphasis on reducing the instruction storage requirement by defining variable length records in the current SUMC documents (References 1 and 2). If this is the case, there should be little reason to have a displacement field less than one-half the word length, except as a tradeoff for base registers. But the instruction set should make every attempt to utilize the displacement field should multiple-length instructions not be implemented.

Scratch-Pad Memory Allocation

The following subparagraphs discuss possible uses of scratch-pad memory (SPM). Four basic approaches are included, and the most productive approach will probably be a combination of the last three, with few exceptions, implementation approaches imply extensions to the current SUMC design. Two major handicaps for using SPM locations for high-speed storage of data or instructions associated with special microprograms are the selection of a SPM location and the selection of the next microinstruction. Both handicaps can be overcome by extensions to the current logic. But to minimize changes to the SUMC, a combination of the second and third approach can be taken along with the chained instruction alternative to the fourth approach. Later, an address control change is discussed with formats (Section 2.3.6.1).

Larger Register Set—Increasing the number of registers (accumulators or index) beyond 16 does not appear necessary if base registers are defined as a separate set, and would require an instruction format change. A few operations such as matrix multiplication (3 x 3) could utilize the extra registers but regardless of the number of registers present, their previous contents usually require a store operation and at times a reload. Furthermore, the major overhead is with the instruction and additional data fetches or the time required to sequence and request mathematical operations and not with the lack of addressable registers.

An additional problem with increasing the number of registers is the increase in task switch overhead caused by the increased numbers of registers to store and load (4 μsec for each additional register per switch).

Multiple Register Sets—Modifications will have to be made to define independent sets of registers in scratch-pad memory (SPM) and have the selection process switch automatically with a task switch. Duplicate sets of instructions would not require a change, but are obviously not the answer. Microprogramming an SPM area switch during task switch will accomplish the same function as a control logic switch, but at best, adds 2/3 μsec for every two words switched.

The SPM-to-SPM microprogrammed switch is 6:1 faster than a generalized main memory switch. The approach is limited to not more than 3 levels, assuming 16 words for an Executive control area. But with more than four base registers involved for each task, the number of levels would have to be cut to two.

High-Speed Data Storage—Assuming that 16 general purpose (accumulators or index) registers were defined and 16 locations were utilized for base registers and Executive control functions, 32 locations would be available for working storage for complex microprogrammed instructions. An instruction can be provided to load and store the data independent of the actual data manipulation instruction to take advantage of the data residence on subsequent instructions.

Loading and storing should be variable-length, multiple-word operations to and from main memory or to and from the general registers. Independent loading does present extra error possibilities, but no worse than requesting the instruction to operate on bad data from main memory.

Using SPM for data storage will increase the number of read only memory (ROM) locations required to define the instruction set since the microprograms are usually more complex and repeat or loop cycles may be difficult to define. Several hardware changes are required to take full advantage SPM for temporary data storage, but the existing design permits many algorithms requiring repeated access to variables and/or constants to be implemented. Constants are especially attractive, since certain algorithms that utilize constant tables are requested repeatedly by a task during the same execution cycle. Algorithms utilizing constant tables could check the residence status of a table and reload only if a task switch has occurred since the last usage or be completely controlled by the Executive.

High-Speed Instruction Storage—As with constant tables, certain instruction sequences are popular to certain tasks and are frequently used during an execution cycle. For instance, a certain conversion or edit sequence is only used normally by the display output routine, but when the program is active the sequence is repeated hundreds of times.

Several changes would have to be made to the SUMC design to permit instruction execution from SPM, but the only major item would be the addition of a route from SPM to the instruction register. Once modified, special operations such as SUMC checkout programming would become easier and specialized instructions could be added at the task level without extensive preplanning to define ROM.

An alternate method requiring no change is to chain the requested instruction to the most frequently requested next instruction. When desirable, a recognizable exceptional state can be used to cause the instruction chain to break. A special requirement is imposed on the placement of microprograms.

This can be accomplished with the SUMC when the initial instruction has the decrement field available. For instance, in a register to register, operation may specify the registers that may be used by either or each chained instruction. The chain may be broken by any microprogram handed the chaining flag. Repeating instructions are also possible using a similar recognizable flag. Neither solves the checkout program requirements.

The SUMC must have the capability to retain the instruction field contents as they are used by one or more microprograms. Each program need not use all of the register fields and each should be able to use the value in a different SPM addressing context.

Immediate data, besides those bits used to request the particular instruction, can be used as intermediate operands. The sign bit of I can be the chaining flag to reduce microprogram detection and reset logic.

2.3.6 Instruction Set

A set of instructions are defined in the following subsections. The instruction set cannot be considered a minimum set but group 1, as defined in Section 2.3.7, is near minimum and does not provide any of the following items:

A. Floating point.
B. Floating point double precision.
C. Fixed point double precision.
D. All specialized functions.

Group 1 does provide many of the functions utilized in Space Station subsystem processing, but lacks significant capability for experiment support processing unless Groups 1, 2, and 5 are implemented. Few instructions listed will assist in the definition of multiprecision operations beyond double.

Section 2.3.6.2 makes tentative register and instruction field assignments for the purpose of instruction definition, but a more optimum assignment requires a more detailed analysis.

2.3.6.1 Format Definitions

The following subsections present a summary of the formats on which the instruction set definition is based. All are to be considered tentative and serve only to demonstrate what can be accomplished with the instruction set and SUMC modifications for extensions discussed in the preceding section. The reference to the protect bit or key in the program counter assumes the approach of minimum SUMC changes with no extensive hardware additions which can accomplish these functions once the hardware unit has been initialized. If the hardware is available, the restrictions on the use of base registers can be relaxed and the format changed accordingly.

Instructions

Each of the currently listed formats (Reference 1) have certain drawbacks when examined with each of the concepts discussed in the preceding section (Figure 2-32, Format 1 and 2). For consistency during program generation, the base register should always be associated with the displacement field since instructions with immediate data can be considered a logical substitute. Other instructions not requiring a full displacement value will still have the capability to address three independent registers.

The accumulator should be the source register for all mass memory storage instructions as well as the target register for all load instructions from scratch or main memory. The index register field is either an index to the main memory reference or specified input data (accumulator or SPM

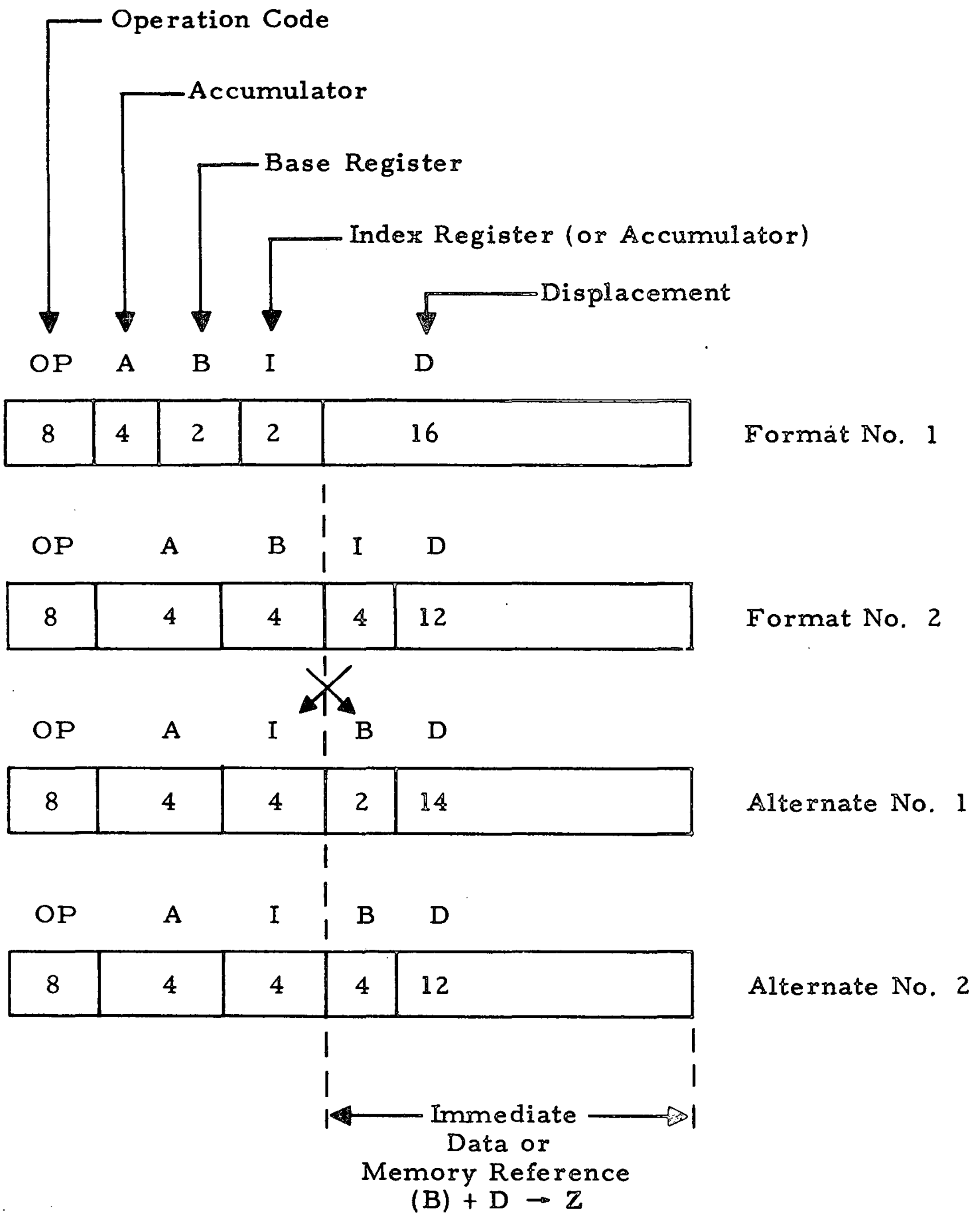


Figure 2-32. Instruction Formats

displacement) for the operation. The base register, if a main memory reference is not involved, may also specify input data for the operation; otherwise, it may be used with the displacement field as data.

The relative size of the base register and displacement field is a trade item because more registers can simplify subroutine parameter passing but increases base value maintenance complexity. The major limiting factor is combining the concepts of multiple registers sets and Executive maintained base registers (i.e., separate from other registers and protected from application programs). Alternate 1 will be more flexible if all SPM utilization concepts remain under hardware control.

Alternates 1 and 2 and Format 2 have a disadvantage if triple accumulator specification is implemented because the decrement field is wasted except for possibly the last six positions. Dual register specification may be less flexible at times; half word data are needed and the formats will permit inclusion of an immediate data value in the computation.

Data

The study effort to define an instruction set has not found any reason to adjust any existing (Reference 1) data formats. The originals and two additional formats are depicted in Figure 2-33, and as a group, define the primary input and output formats for the instructions defined in a subsequent section.

The decimal format is used as an intermediate step between a standard eight-bit character string required for displays and controls and the formats required for arithmetic computation. The four-bit code is ideal for the SUMC shift (multiplexor) logic and permits easy packing while editing eight-bit codes. The eight-bit code should be the standard, as shown in the character format. The intermediate four-bit code may be bypassed when converting to decimal code since the final process involves a merge of a valid string of digits to a display character format.

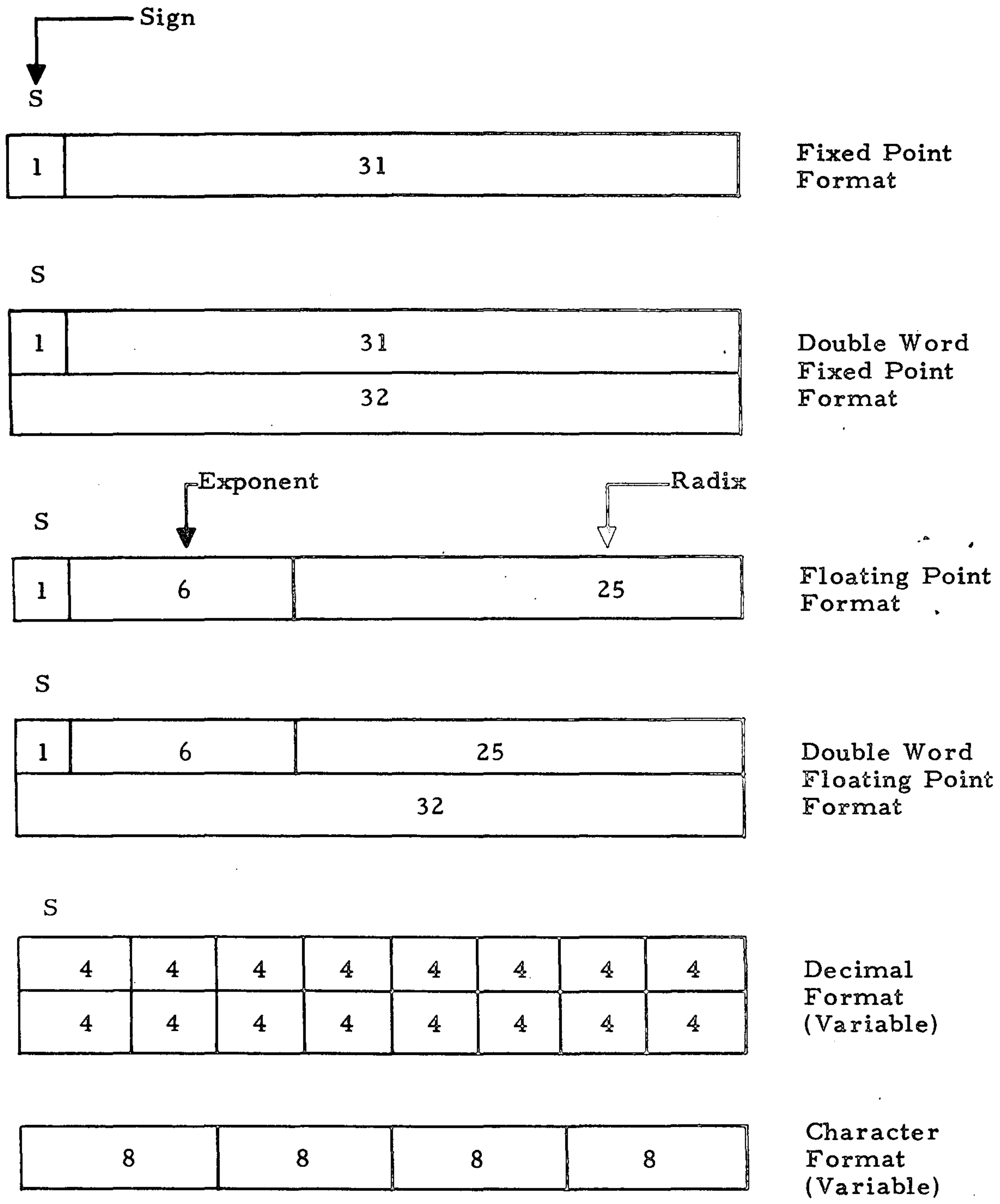


Figure 2-33. Data Formats

Input/Output

All instructions associated with the operation of the input/output controller reference a normal 32-bit main memory word format without regard to its content. Therefore, there are no special format considerations for I/O instruction definition. The data bus word format (Reference 4) is presented in Figure 2-34 to support subsequent instruction types and examples of instruction utilization and the need for half-word operands.

Internal Control

The Executive program specification will define the control blocks that may permanently or temporarily reside in scratch-pad memory (SPM). A preliminary examination indicates that at least the following types of information will reside in SPM in addition to the registers described earlier.

A.. Program Counter which contains the memory address of the currently executing task.

B. Execution Control which contains the current program base, the storage protection key or memory block upper limit and program state indicators (i.e., privileged, interruptable, relocatable, etc.).

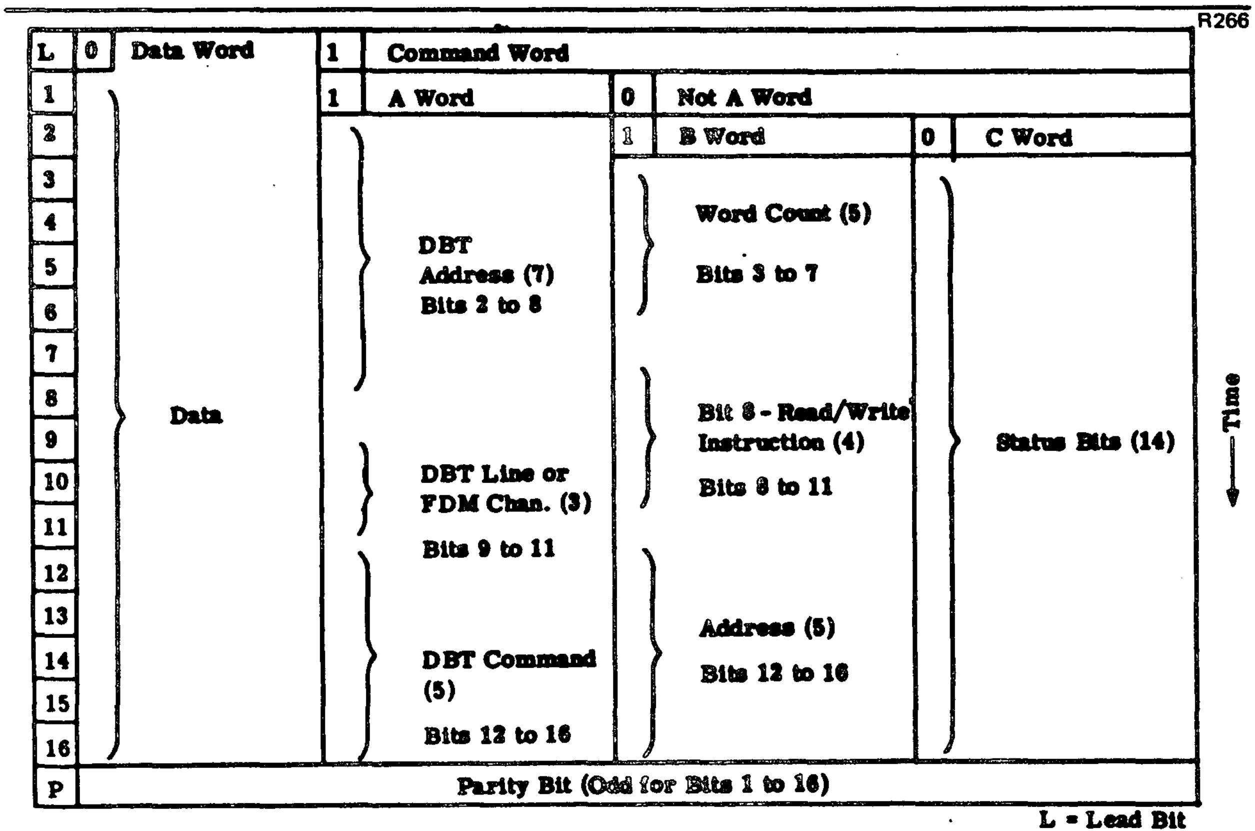


Figure 2-34. Data Bus Word Formats

L = Lead Bit
P = Parity Bit

C. Program Control which contains indicator bits or status associated with the execution of arithmetic or special instructions and Executive options, program or task entry for exception processing, and mask flags to inhibit exceptional processing.

D. Task Control which contains a reference to the program structure maintained by the Executive to determine current responsibility should an abnormality occur.

E. Executive Control which contains information similar to program control but available only to the Executive.

F. Base registers which contain the memory reference address as defined by the program, storage protection key, and mass memory state indicators (i.e., read only, privileged, etc.).

The above information is tentative but is depicted in Figure 2-35 to provide support information for the instructions defined in a subsequent paragraph.

Scratch-Pad Memory

Figure 2-36 depicts the four SPM utilization concepts discussed in Section 2.3.5.4, but does not show the possible combinations. Concept three and four are similar and their implementation depends on the feasibility of implementing the hardware changes required to gain significant benefits. Of course, concept two and four could be combined without major changes and are the basis for defining the instruction set (Section 2.3.6.2). The instruction format implied is number 1 or alternate number 1; otherwise, all base registers could not be contained in the internal control area.

The format requires special instructions for the Executive to manipulate SPM or modifications to the SPM address control logic. For instance, bits one and two can be used as '00' while defining instructions but forced to 'X1' while not executing in the privileged or Executive state. Then, if privileged instructions were defined to control the first or both bits by establishing secondary state flags, complete flexibility to address any SPM location could be given to the Executive. A logical inclusive Or of the flags and address bits one and two provide the necessary capability without data movement.

Section 2.3.6.2 includes special instructions for the Executive to manipulate SPM, should the address control changes not be made.

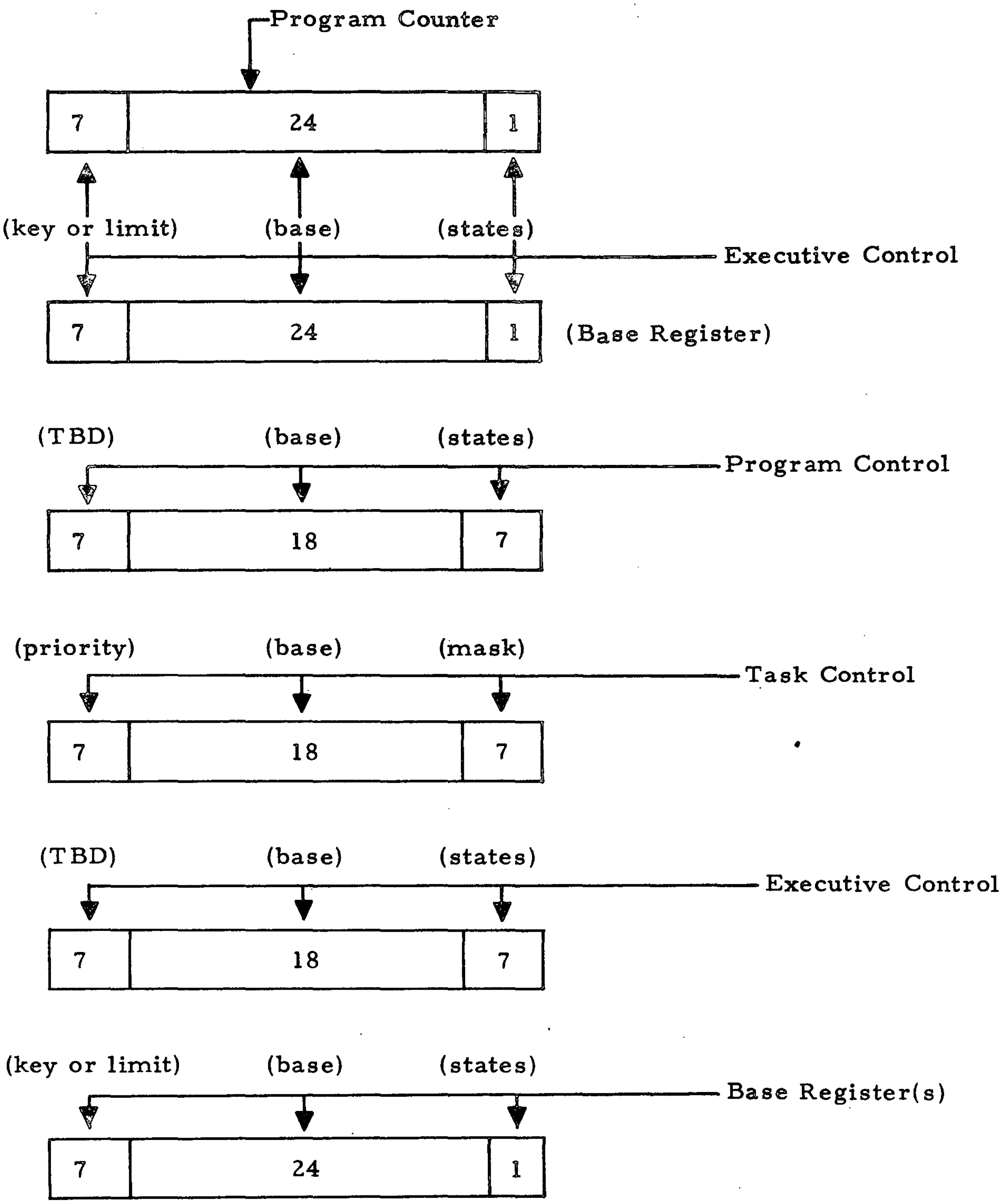


Figure 2-35. Internal Control Blocks

(SPM)	Concept 1	Concept 2	Concept 3	Concept 4
0 15	Internal Control and Temporary Storage	Internal Control and Base Registers	← (Same)	← (Same)
16 31	Accumulators	Accumulators and Index Registers (Section 1)	← (Same)	← (Same)
32 47	Index Registers	↑ (Section 2, Same)	Program Segments, Constants and Working Storage (Overlays)	Constants and Working Storage (Overlays)
48 63	Base Registers	↑ (Section 3, Same)		

Figure 2-36. Scratch-Pad Memory Utilization Concepts

2.3.6.2 Instruction Definition

Even though a variable length instruction concept has not been recommended as a possible design change for SUMC, an attempt has been made to maximize the utility of the B and D fields when they are not required for addressing. The format definition placing the two fields adajcent permits ready access to an additional half-word value during any register-to-register operation. No additional time is required for this operation assuming the current Fetch Microprogram (Reference 1).

The effective address of any main memory location is determined by ((D) + (X) + (B)) and is referred to as Z. Half word memory locations are designated as either Z_l or Z_r. The notations to be used in the following paragraphs include:

( ):	is read as the contents of.
A:	location referenced by accumulator field.
X:	location referenced by index register field.
B:	location referenced by base register field.
D:	displacement field of the instruction.
R_x:	is a specific general purpose register (0, 1, etc.)
I:	immediate data or (B, D) taken as data.
Z:	effective memory location((D + (X)) + (B).
0 or 1 (subscript):	a given register of an even/odd pair.
1 or 5 (subscript):	left/right half of 32-bit word.

<u>SPM Content Control</u>

The following instructions have been defined to include the basic operation for the effective utilization of scratch-pad memory. Base registers are assumed to be independent of the set of general registers which can be specified in either the A or X fields by privileged or nonprivileged instructions. Index register fields equal to zero normally do not select any register and a zero is used in the described load or store operation. Memory location Z is always indexed with X unless otherwise noted.

The instructions that reference half words should have their operation codes modified during program generation by an assembler or compiler. All other load and store SPM operations have been simplified by utilizing the same instructions for fixed point or floating point formats.

Base registers must be assigned to the higher numbered locations of a 16-word SPM area to permit their being addressed by A and X fields in the instruction if the B field is only two or three bits. If all protection features are implemented with keys controlled by hardware units, base registers can be common with general registers. With either approach, the number of base registers can be increased to eight and remain separate. But with key protection, generalized load, store, and manipulation can be permitted and new instructions defined.

General Registers—The following set of instructions is based on the assumption that there are 16 SPM locations being utilized as either accumulators or index registers. Each instruction is nonprivileged and may be executed by any program and all instruction fields may be utilized.

A. Load or Store (L, ST), which transfers a full word of data between any accumulator and addressable memory locations [(Z) → A or (A) → Z].

B. Load or Store Left or Right Half (LLH, LRH, STLH, STRH), which transfers either the right or left half of an addressable memory location and any accumulator. The load is to the right half of the specified accumulator, and the sign is extended in the left half [$(Z_1) \rightarrow A$, $(Z_r) \rightarrow A$]. On a store operation, the left half of the accumulator is ignored and the right half replaces one of the addressable memory locations [$(A_r) \rightarrow Z_1$ or $(A_r) \rightarrow Z_r$].

C. Load or Store Double (LD, STD), which transfers two full words of data between any accumulator pair and any two addressable (adjacent) memory locations. The even register is specified [$(Z, Z + 1 \rightarrow A_0, A_1$, or $(A_0, A_1) \rightarrow Z, Z + 1$].

D. Load or Store Multiple (LM, STM), which transfers a consecutive set of full words between any addressable memory vector and a set of registers. This instruction is used to store reduce the overhead of loading or storing the set of general purpose registers [$(Z_0, \ldots Z_j) \rightarrow R_i, \ldots R_{15}$ or $(R_i, \ldots, R_{15}) \rightarrow Z_0, \ldots Z_j$], where (i) is the low order three bits of I and (j) is (15 - i). The instruction is implemented with chained logic or the SUMC modifications.

E. Load Absolute (LA, FLA) of a single precision fixed point or floating point memory location Z and leave the result in A.

F. Load Immediate/Reverse Subtract (LIRS) is accomplished similar to Load Right Half (LRH) except the operand originates from the right half of the instruction if the index field is zero (X = 0). Otherwise, the instruction accomplishes a subtract register (SR) type function but utilizes the reverse subtract operation of the ALU (I - (X) → A).

G. Load Immediate Address (LIA) is similar to the load (L) instruction except that A is loaded with the address reference and not the data (Z → A).

H. Load Immediate (LI) is similar to a Load Right Half (LRH) except the data is taken from the right half of the instruction in the form of immediate data and replaces the contents of A and clears X if $X \neq 0$ and $X \neq A$.

I. Load Byte (LBD, LBI) loads the bytes (eight-bit) referenced by immediate data and leaves the bytes right adjusted in A with remaining bytes set to zero. The byte address is determined by $Z\,[D + (X,\ X \neq 0)]$ and the word address is Z/4 with the remainder to select the byte position. Indirect (LBI) is accomplished by loading (Z) and then accomplishing the load byte operation with (Z) as a byte reference.

Privileged Areas—The following instructions are defined to control the contents of all scratch-pad memory locations. The instructions are defined to transfer data to or from locations outside the general register areas. The source or destination associated with the data transfer may be main memory or general register locations.

The first two instructions are nonprivileged and may be executed by any program; all others are defined for Executive program use. If the number of base registers is increased from four or if the SPM utilization concept is modified, the instructions must be reviewed for possible change. Except for the last instruction defined (LEB), the special privileged instructions would not be necessary with special SPM address control logic for the Executive to manipulate bits one and two of the SPM address.

A. Load Program Control (LPC) is a nonprivileged instruction used to modify information in the Executive-controlled area of scratch-pad memory. The instruction loads a mask corresponding to the program control states or state indicators defined during instruction implementation and an address reference for processing of states not masked. The address reference may be any addressable program location, and the mask is selected from the least significant seven bits of the general register specified in the A field $[(B) + D, (A)_{25-31} \rightarrow PC)]$.

B. Load Base or Load Base from Register or Load Base from Base (LB, LBR, LBB) is a nonprivileged instruction to define the portion of a base register not controlled by the Executive. Otherwise, the instruction is implemented similar to a Load (L) or Load Register (LR) with immediate automatically added. The Executive controlled portion of the new base (C_b) is derived from the reference base register or from the base corresponding to the program counter ($Z \vee C_b \rightarrow A_b$, $(A) + I \vee C_b \rightarrow A_b$, $(X_b) + I \rightarrow A_b$).

C. Load or Store Privileged area 1, 2, or 3 (LP1, STP1, LP2, STP2, LP3, STP3) is specified and operates as a Load Multiple (LM) or Store Multiple (STM), except one of the three remaining areas of SPM is involved. The instruction LP1 can act as a branch type instruction unless the location corresponding to the program counter is preplanned or the next location following the LP1 instruction (Z, $Z + 1, \ldots \rightarrow$ area (i) or area (i) $\rightarrow Z + Z + 1 \ldots,$).

D. Load and Store Registers with Privileged area 1, 2, or 3 transferred (LRP1, SRP1, LRP2, SRP2, LRP3, SRP3) concurrently with 16 memory words (Z, $Z + 1, \ldots, Z + 15$) either to or from general registers. Concurrent with the operation, either a transfer of the previous contents of the general register to a specific privileged area (LRP1, LRP2, LRP3) or the general registers are reloaded from a specific privileged area (SRP1, SRP2, SRP3). The instruction will speed normal task switching (Z, Z + 1, . . . , Z + 15 → general registers → area (i) or area (i) general registers → Z, Z + 1, . . . , Z + 15).

E. Exchange Privileged with Registers or Load Registers to Privileged or Privileged to Registers (EPR, LRTP, LPTR) accomplish any combination of SPM to SPM multiple-pair loads or content exchanges between the general purpose registers and privileged area I_p. With each instruction, both receiving and sending locations are designated by I_r and the number of pairs by I_n. The implementation will require chaining logic (Section 2.3.5.4)or SUMC modifications. The instructions are utilized by the Executive to manipulate privileged areas without going through main memory (area (I) → (R_I), (R_I) → area (I), or (R_I) → area (I).

F. Load Executive and Branch (LEB) is similar to the load program control instruction (LPC), but two Executive control words are loaded in privileged area 1. Other differences are that the contents of the locations Z, . . . , Z + 3 are used to load the two control words, base registers zero, and to reset the program counter for a return to a privileged or privileged program.

Arithmetic Instructions

All arithmetic operations are carried out in floating point or fixed point with input and output formats as defined in the preceding section. Status flags should be defined corresponding to the exceptional conditions defined during the implementation of all instruction types. Included in the set should be any item that may not require immediate action or where processing time can be saved by not verifying or testing status after each operation. The status flags remain set until tested or stored.

Fixed Point—Fixed-point arithmetic operations are performed in 2's complement form for either single or double precision. Half-word and partial-word parameters developed by masking are operated on with full-word logic. Status flags include both underflow and overflow and the instructions are defined as:

1. Add or Add Double (A, AD), which adds the contents of any addressable memory location or adjacent locations to the register

or register pair designated by the A field [(A) + (Z) → A, $(A_0, A_1) + (Z, Z + 1) \rightarrow A_0, A_1$].

2. Add Left or Right Half (ALH, ARH) which adds either the left or right half of any addressable memory location to the right portion of the register designated by the A field [$(A) + (Z_1) \rightarrow A$ or $((A) + (Z_r) \rightarrow A$]. The sign of the half word is extended prior to the addition.

3. Add Register (AR) under mask is the addition of the register specified by the X field after a Logical And has been accomplished with immediate data $(A + (X) \wedge I \rightarrow (A))$. Register zero is selected if $X = 0$.

4. Add Absolute of a single or double precision (AA, AAD) operation takes the absolute value of Z or Z and Z + 1 and adds the results to A or A_0A_1 $(|Z| + (A) \rightarrow A$ or $|Z, Z + 1| + (A_0, A_1) \rightarrow A_0, A_1)$.

5. Subtract instructions (S, SD, SLH, SRH, SR) include the same basic operations and parameter designations as the corresponding add instructions. With immediate data, a Logical And $(I \wedge (X))$ determined as the first step.

6. Multiply instructions (M, MLH, MRH, MR) include the same basic operations and designations as the corresponding add instructions. With immediate data, a Logical And $(I \wedge (X))$ is determined as the first step. The exception is that on all full or double word operations, the A field must designate the even register of an adjacent pair. On single precision operand is selected from A_1. Half-word multiplications may designate any register and the 32 most significant bits replace the product. Multiply (M) or Multiply Register (MR).

7. Divide instructions (D, DLH, DRH, DR) include the same basic operations and designations as the corresponding add instructions. With immediate data, the arithmetic sum is determined as first determined and A must designate an even register.

Floating-Point—Floating-point operations are performed on single or double word formats as with fixed-point. Underflow or overflow flags may be set as well as abnormal exponent status. On register type, immediate data may be used to reference systems or Executive controlled data or to chain the instruction to other instructions (see Section 2.3.6.2). Either approach may be used to simplify implementation or reduce Read Only Memory requirements.

A. Floating-Point Add and Floating-Point Add Double (FA, FAD) are specified as with the corresponding fixed point instructions (A, AD) except all parameters are expected to be in a normalized floating point form. The sum is always normalized and replaces the specified register or register pair [(A) + (Z) → A or (A_0, A_1) + (Z, Z + 1) → A_0, A_1].

B. Floating-Point Subtract or Floating-Point Subtract Double (FS, FSD) instructions include the same basic operations and parameter designations as the corresponding floating point add instructions.

C. Floating-Point Multiply or Floating-Point Multiple Double (FP, FMD) instructions include the same basic operations and parameter designations as the corresponding floating point add instructions.

D. Floating-Point Divide or Floating-Point Divide Double (FD, FDD) instructions include the same basic operations and parameter designations as the corresponding floating point add operations.

E. Floating Point Add Absolute of a single or double precision value (FAA, FAAD) takes the absolute value of Z or Z and Z + 1 and adds the results to A or A_0A_1.

F. Floating Point Add, Subtract, Multiply or Divide Registers (FAR, FSR, FMR, FDR) is similar to the corresponding memory operations except that the data is selected as (X).

Format Conversion—The instructions listed are defined to speed processing for the Displays and Controls Subsystem. The conversion sequence supported is from packed decimal resulting from an edit operation on keyboard input to a floating-point form and then if required, to an explicit scaled fixed point format. Fixed-point arithmetic operations may maintain an explicit or implicit scaling. The reverse process is supported by converting an explicitly scaled fixed-point number to floating point and then converting the floating point format to a character string for subsequent merging to any display format.

Immediate data can be used to reference system or Executive controlled data or to chain the instruction to other instructions (Section 2.3.5.4). Either approach may be used to simplify implementation and reduce read only memory requirements.

A. Scaled to Floating Point (STF, STFD) to transform a fixed point single or double precision format to the normalized equivalent floating point format. The output or floating point number replaces the input or fixed point number (A or A_0, A_1). The input is a signed number whose scale is specified as the effective value of the displacement field plus an optional register value ($X \neq 0$).

B. Floating Point to Scaled (FTS, FTSD) to transform a floating point single or double precision format to the equivalent fixed point format. The output or fixed point number replaces the input or floating point number (A_0 or A_0, A_1). The output is a signed number whose scale is specified as the effective value of the displacement field plus an optional register value ($X \neq 0$).

C. Decimal to Floating Point (DTF, DTFD) converts a string of decimal digits to a single or double precision floating point number. The Display and Control Subsystem character strings must be scanned and valid strings of four bit digits (0000-1001) packed in to general register A_0 and A_1. The digits are right adjusted and the left most bit of R_0 specifies the sign and the least significant bits specify the number of valid digits. R_1 contains the power of ten in 2's complement binary. Z is reserved to reference the logarithm table for radix conversion. The results replace A or A_0, A_1.

D. Floating Point to Decimal (FTD, FTDD) converts a single or double precision floating point number to a valid character string of eight-bit characters whose least significant four bits are (0000-1001). The floating point number is specified by A and double precision must be in A_0 A_1. The sign of the number replaces the sign bit of R_0 and the correct power of ten is placed in R_1 in 2's complement binary. Z is reserved to reference the logarithm table for radix conversion. The character string is stored as eight bit characters starting at the location specified in R_0 and is packed four per word. The most significant bits of the output are always 0000 as is the unused character positions of the last word.

Bit Manipulation

The following instructions are included to perform general operations on all data bit strings up to and including a length of 64. They are normally accomplished in a single register or even/odd register pair. Load Register type does not have to hold to this convention. All instructions involving main memory may specify an index register ($X \neq 0$).

Shift Operations — The following shift operation should be implemented but the Shift and Count Double and both Rotate instructions may be omitted to minimize the instruction set. The amount to shift or rotate should be the sum of the displacement field and the register contents specified by the index field ($X \neq 0$) to remove requirements for instruction modification.

A. Shift Right or Left Logical either a single or double register (SRL, SLL, SRLD, SLLD). A left shift results in zeros entering the right portion of the register and the loss of any bits shifted beyond the left end. A right shift results in zero bits shifted beyond the right end.

B. Shift Right or Left Arithmetic either a single or double register (SRA,.SLA, SRAD, SLAD). The operation is identical to the corresponding logical shifts except that the sign bit is extended on right shifts and an overflow condition is detected on a left shift. Arithmetic left shifts can be omitted to minimize the instruction set.

C. Shift Left and Count Single or Double (SLC, SLCD) is accomplished similar to a ShiftLeft Arithmetic Single or Double (SLA, SLAD), except that a count is not specified and the shift continues until the left most bit (sign) does not match the next bit to the right (most significant bit). The register shifted is A or A_0, A_1 and the number of shifts added to the displacement field replace the contents of X.

D. Rotate Left or Right Logical (RLL, RRL) is accomplished similar to the corresponding shift left or right logical except during shifting bits are not lost but are entered in the opposite end of the register. The count is specified in D and (X) is rotated and replaces (A).

Logical Operations — The following logical operations do not have special condition indicators assigned and each is a basic function of the SUMC arithmetic logical unit (ALU). Load or store type operations permit indexing (X) within memory and register to register operations permit defining multiple level operations utilizing immediate data or instructions. The immediate data usage defined for the instructions can be redefined to accomplish instruction chaining (Section 2.3.6.2). At this time, it appears that the masking operation will be a requirement.

A. Load Logical And, Inclusive Or, and Exclusive Or (LAN, LIR, LER) uses the contents of the specified memory location (Z) and operates on the specified register (A). The results are placed in the odd register A_1 and the contents of Z are not modified [$(Z) \wedge (A) \rightarrow A_1$, $(Z) \vee (A) \rightarrow A_1$, $(Z) + (A) \rightarrow A_1$]. The specified register is not modified unless $A = A_1$.

B. Store Logical And, Inclusive Or, and Exclusive Or (STAN, STIR, STER) is specified and carried out identical to the corresponding load logical operations except that the results are returned to the memory location Z. The contents of A are not modified [$(Z) \wedge (A) \rightarrow Z$, $(Z) \vee (A) \rightarrow Z$, $(Z) + (A) \rightarrow Z$].

C. Load Register with immediate Logical And, Inclusive Or, and Exclusive Or (LRAN, LRIR, LRER) is similar to a load immediate in that a 32-bit operand is generated from the right portion of the instruction. The named logical operations are only limited by the expansion algorithm for immediate data [$(X) \wedge I \rightarrow A$, $(X) \vee I \rightarrow A$, $(X) + I \rightarrow A$]. The contents of X are not modified unless X = A.

Branching

Two types of instructions have been defined for nonprivileged programs to alter the normal execution of sequential instructions. A third type occurs but without direct instruction execution by a nonprivileged program. The latter type includes interrupts associated with I/O and internal conditions such as storage protection and instruction protection detected by internal or microprogram controlled logic. The fourth type also includes microprogram controlled branches associated with the Executive and program status condition that may be selectively trapped following a Load Program Control instruction. The two instruction types are associated with conditional skips or branches and unconditional branches.

Conditional Skips or Branches—The skipping of sequential instructions are always relative to the established program counter. Skips are more restrictive since addressing is always relative to the base register associated with the program counter. Branches can specify any base register.

A. Compare a single or double precision number in either fixed or floating point (C, CD, FC, FCD). The operation compares the arithmetic value contained in A or A_0 and A_1 against a value contained in Z or Z and Z + 1. The relationship is stored as three status indicators along with other status conditions (i. e., overflow, underflow, etc.). The status bits are reset with each Compare or Test Status.

B. Test Indicators and Skip (TIS) uses immediate data (I) to test the logical states of one or more status indicators. If any corresponding bit is set, a skip will occur. The indicators are always reset with the instruction and if $A \neq 0$, the results of the logical and operation replace the contents of A.

C. Branch on Zero, Not Zero, Positive, Not Positive, Negative, Not Negative indicators set or not set (BZ, BNZ, BP, BNP, BN, BNN). The operation is similar to combinations of compare and skip type instructions, except that no compare operand is required [(A): zero].

D. Decrement Count Branch Positive (DCBP) decrements A by one and branches to Z if the result is not negative [(A) - 1 → A; if negative P.C. + 1 → PC].

E. Increment Count and Branch Negative (ICBL) increments A_0 by one and branches to Z if $(A_0) < (A_1)$; otherwise, goes to the next instruction (P. C. + 1 → P. C.).

F. Decrement by Register and Branch Positive (DRBP) decrements A_0 by X and branches if (A_0) is positive; otherwise, goes to the next instruction (P. C. + 1 → P. C.).

G. Increment by Register and Branch Low (IRBL) increments A_0 by X and branches if $(A_0) < (A_1)$; otherwise, goes to the next instruction (P. C. + 1 → P. C.).

H. Test Bits (TBAN, TBER) to test for bits in Z that match A with a Logical And (TBAN) or to determine bits positions that do not match (TBER). If the results are zero, execution goes to the next instruction; if not, the results replace A_1, and the next instruction is skipped.

Unconditional Branches—Unconditional branches can specify any base register like the previously defined conditional branch instructions. A program task may be given access to a set of base references or instruction areas but without being given storage privileges. The task or program executing as a part of the task may branch to reentrant programs within those areas without explicit requests to the Executive. If required, the Executive can determine the responsible task by examing task control information held in SPM even with several tasks sharing reentrant program elements.

A. Branch (B) unconditionally to the specified location (Z) incremented by (X) if $X \neq 0$. The possibilities include the reference to any established base register, a zero or a positive displacement value, and a zero or nonzero index value $[(X, X \neq 0) + D + (B) \rightarrow$ program counter].

B. Branch and define Linkage for Return under current task or executive control (BLR, BLRE). Both are specified similar to the Branch (B) instruction, except that current program counter reference position is stored in the register indicated by A. With the local instructions, the reference position is only a displacement value and may be used with a Branch instruction to return control, branch to (X). With the global instruction, the same operation is performed, except that a global reference must have been preestablished by the loading of the base register to be referenced. A linkage to the Executive is required to adhere to storage and instruction protection features; all other branch types must be to areas assigned by the Executive as controlled by the key or address limit concept.

Special Purpose

The following instructions accomplish specific functions such as input/output control. Others are included to facilitate implementation of guidance, navigation, and control; display and controls, and experiment processing algorithms. Any of the latter type may be omitted to minimize the instruction set.

Input/Output—There are two types of I/O instructions, those directing immediate execution and those initiating asynchronous operation of an attached I/O control unit. The latter is usually implemented as an interrupt or implicit branch operation to an assignable instruction location upon CPU recognition of the requested operation having completed. The Executive program will be responsible for maintaining all addresses associated with I/O interrupts including SPM flags to direct the microprograms which are defined to interface with the I/O control unit. Only three instructions have been defined for I/O, two of which are immediate and do not utilize interrupt logic. Each microprogram defining the instructions must recognize a failure on the part of the IOC unit to return associated status information and terminate the particular instruction. Normal execution which results in status information being returned should be implemented to skip the next instruction.

A. Begin Input/Output (BIO) instructs the CPU to build and output a control word to the attached Input/Output Control (IOC). The memory address is combined with the channel identification contained in A_0. Any status information returned by the IOC unit is placed in A_1.

B. Halt Input/Output (HIO) is implemented similar to the Begin Input/Output instruction (BIO) except that the memory address read does not have meaning to the IOC unit. The specific channel is designated in A_0 and the status information is returned in A_1.

C. Test Input/Output (TIO) is implemented similarly to the BIO instruction, but like the HIO, the operation does not initiate asynchronous operation of the IOC unit and the microprograms delay until the status information is returned. A time-out must be included to recognize IOC unit failure.

Multiprocessor—There are three instructions that reduce the task of expanding from a simplex to a multiprocessor configuration. The three should be implemented so that they may be utilized even when the multiprocessor units are configured as a simplex system. The definition of the first instruction, Load Identity, provides information to the Executive or maintenance programs to reduce the logic required to accomplish fault isolation, automatic reconfiguration, and crew coordinated maintenance. The identification may be a unique value placed in the Read Only Memory (ROM) of each CPU or may be an internal or microprogrammed sequence to establish uniqueness by determining physical interconnection relationships among the units.

The other two instructions are utilized to detect conflicts among the CPU's through lock flags stored in shared main memory locations and to reduce memory activity from a CPU looping and waiting for a lock to be removed. Both the Test and Delay instructions may be executed by any program. The three instructions are defined similar to equivalent instructions in an existing multiprocessor (Reference 3). The Delay instruction permits any program to gain access to a double word real-time clock maintained in SPM. Such an instruction feature is desirable even in a simplex system.

A. Load Identity (LID) which instructs the CPU to determine unique identification and place the value in scratch-pad memory (A). LID is a privileged instruction.

B. Test/Set and Skip (TSS) is an instruction that takes the bits of A and if any corresponding bits of Z are set, the instruction is terminated, Z is not modified, and the next instruction is executed. If no corresponding bits were set, a logical inclusive or places the bits of A into Z and the next instruction is skipped.

C. Delay and Store Time (DST) is an instruction requesting the CPU to loop in a microprogram for a variable number of microcycles based on the immediate data value. The delay interval is N + k I where N is the initial overhead for execution of the instruction and k is the overhead for testing for interrupts, checking interrupt masks, and for microprogram loop control. An unmasked interrupt will terminate the delay and return to the Executive with the program counter indicating completion of the Delay instruction. If X ≠ 0, the value of the specified register is added to the immediate data to determine the time to delay. If A ≠ 0, the current time maintained in SPM replaces the contents of A_0 and A_1.

Arithmetic and Programmed Arithmetic—As with other arithmetic operations, special indicator bits may be assigned to report abnormal condition with a program controlled branch or by the program testing the flag with a conditional branch instruction. Certain instructions that are normally register to register operations can utilize the X, B, and D fields to contain other parameters that are required by the microprogram to accomplish the requested function. The parameters may be constants, variable, or a vector to either constants or variables.

If parameters are not required, the instructions may be designed to load X (X ≠ 0) either from immediate (I) or indirect ((B) + D) data, be defined to chain one microprogram instruction to a second microprogram, or be used to control total or partial microprogram repetitions. The microprograms that are involved in chained operations should have interfaces defined such that individual microprograms can be executed as independent instructions (different operation codes). The following instructions have this requirement and Read Only Memory requirements can be reduced if other complex instructions can be implemented in a like manner.

A. Programmed Sine or Cosine (PS, PC, FPS, FPC) transforms a fixed point or floating point single precision angle A expressed in Pirads to the principle angle A required to accomplish sine or cosine approximation by polynomial approximation. Low-order bits of D reference the appropriate polynomial expansion microprogram. Other bits become pointers to any constants defined in Executive controlled memory to minimize microprogram logic. The sign bit of D is the chain control bit. X_0 (exponent) is initialized to the standard number of terms for single precision sine or cosine approximations. X_1 (exponent) is set to 0, 1, 2, or 3 to reflect the determined quadrant. The instruction is chained to a PE or FEF.

B. Polynomial Expansion (PE, PEF) fixed or floating point uses the vector Z and variable (X) in the expansion process and X + 1 contains the number of terms. The result replaces the contents of A and for accuracy and efficiency, the expansion should be: $\{[(Z_0 A + Z\ \ ) A + Z_3] A + \ldots Z_{n-1}\}$.

C. Programmed Arctangent (PAT, FPAT) initially transforms a fixed point or floating point arctangent (absolute value) arguments [numerator (X_0), denominator (X_1)] in single precision to a single value for subsequent approximation of the principal arctangent by polynominal approximation. Immediate data is used to control the programmed sequence, and the results is placed in the general register specified in the A field following execution of a PE or PEF.

D. Programmed Square Root (PSQ, PSQD, PFSQ, PFSQD) determines the root of a single or double precision fixed-point or floating point of positive value specified by X or $X_0 X_1$ and the result is stored in A. A negative value will result in program control indicator set operation. Either instruction that must be implemented as chained programs requires the immediate field for control.

E. Vector Dot Product (VDP3, FDP3) of two single precision vectors (dimension 3) in fixed point or floating point and store the results in A or A_0 A_1. In fixed point, the additions are carried out in double precision. The reference location of one vector is Z and the reference of the second must be preloaded in base register B_X.

F. Vector Cross Product (VCP3, FVCP3) of two single precision vectors (dimension 3) in fixed point or floating point and store the results Z, Z + 1, Z + Z. The reference addresses of the two vectors must be preloaded in base registers specified by A, that is in B_{A0} and B_{A1}.

G. Multiply Matrix-Vector (FMMV3) to multiply a matrix times a vector in floating point (dimension 3) and store the results in Z, Z + 1, Z + 2. The reference address must be preloaded in base registers specified by A (B_{A0} and B_{A1}).

Repetitive Operations—Instructions that must be repeated to accomplish a particular function can be implemented as microprogrammed instructions that are repetitive where the number of operations are specified by the decrement and/or a register value. The next instruction listed is an alternate approach to accomplishing repetitive operations and acts much like a fixed "DO" loop in Fortran which contains a single instruction.

If the latter approach can be efficiently implemented, several previously listed instructions can gain extra significance, especially operations from memory to register or from register to memory. Several other instructions are listed below that can be implemented with self-controlled repetitive operation or by a loop commanding instruction (IBRN).

Unless efficiency cannot be achieved, several of the fixed length repetitive instructions listed in the previous section can be implemented similar to the following "increment" type instructions.

A. Increment Base Repeat Next (IBRN) accomplishes a similar operation as a Decrement Count and Branch Positives following the next instruction, if the branch reference points back one location. The difference is that the count is contained in the displacement field and the overhead for instruction fetch is reduced. The base register specified by A is incremented by one as the repeated operation is carried out. The implementation impact on other instructions and the overhead of this instruction have not been examined in depth.

B. Increment and Limit Check (ILC, IBLC, IFLC) compare the value referenced by B_{A0} for an in-limit condition as determined by two limit values referenced by B_{A1}. The instruction is repeated the number of times specified by D + (X, if $X \neq 0$) and (B_{A1}) is incremented. If an out-of-limit condition exists, the instruction repetition is terminated, and the next instruction is skipped. If IBRN is implemented, Increment and Limit Check can be defined as a conditional branch. The instructions operate on fixed-point data either full word (ILC) or bytes (IBLC) and floating point (IFLC). With bytes, four are tested with each repetition and if an out-of-limit condition occurs, the pattern replaces the contents of general register A.

C. Increment Test (ITAN, ITER) to test bit patterns of two vectors similar to the test bits instructions (TBAN, TBER). Base registers B_A and B_X must be preëstablished, and following each test, (B_X) is incremented by one. If the result of any test is non-zero, the result is stored in the general register specified by A. Immediate data is utilized to control the repetition and increment B_{A1}. If IBRN is implemented, the instruction can be defined as a conditional branch.

D. Increment and Dot Product (IFDP, IMDP, IBDP) determines the vector dot product of two vectors referenced by B_{A0} and B_{A1}. The length of the vectors are specified as immediate data [D + (X, if $X \neq 0$)]. The parameters are floating point (IFDP), bytes (IBDP) or a combination (IMDP, B_{A1} are floating point and B_{A0} are packed bytes). The product of bytes are maintained in fixed point (A) and floating point or mixed are accumulated in floating point. Vector length, with mixed parameters (B_{A0}) is 1/4 and with bytes, both are 1/4 of the specified repetition cycle.

E. Increment and Sum Bytes (ISB) determines the sum of a packed string of bytes in fixed point and the result is accumulated in A. Immediate data [D + (X, $X \neq 0$] specifies 1/4 the vector length.

Configuration Control—The details of the following instruction are dependent upon the hardware implementation for controlling the set of units that make up the simplex of multiprocessor SUMC. Four instructions are listed to to show the relationship to the SUMC instruction format and other instructions. For more details, reference the multiprocessor system description.

A. Load Configuration Control Unit (LCCU, LCCE) is defined for the Executive to load the configuration control of the CPU executing the LCCU. The Executive, depending on the relative system control states, can cause the loading of the configuration control unit of another unit (LCCE). Subfield usage and content is defined in the multiprocessor system description.

B. Store Configuration Control (SCCU) is similar to the load instructions, but of course, accomplishes the opposite function.

2.3.7 Instruction Set Application

The instructions defined in the preceding section have been listed in 2.3.7.1 and assigned a priority index to define a logical buildup sequence. The sequence may be used as implementation steps to insure consistent specification or breadboard evaluation. Figure 2-37 presents an alternative for SPM allocation should all or almost all of the instructions be defined and the number of base registers prove to be too limiting. Examples of typical processing functions are presented in Section 2.3.7.2 along with instructions required to demonstrate capability. The instruction examples assume the alternate SPM allocation.

Assignment	SPM	Normal Usage
Privileged Area No. 1	Executive Control	
	Base Registers (8)	(Inactive Task)
Privileged Area No. 2	Work Area	Executive or Microprograms
	Base Registers (8)	(Active Task)
Privileged Area No. 3	Stack or Extra Work Area	(Inactive Task) GPR's
Nonprivileged	General Purpose Registers	(Active Task) GPR's

Figure 2-37. SPM Allocation

2.3.7.1 Instruction Summary

General Register Content Control

Note: (*) indicates privileged instruction

Designation	Index	Name
L	1	Load
ST	1	Store
LLH	5	Load left half
STLH	5	Store left half
LRH	5	Load right half
STRH	5	Store right half
LD	3	Load double
STD	3	Store double
LM	5	Load multiple
STM	5	Store multiple
LI	1	Load immediate
LIRS	1	Load immediate/reverse subtract
LIA	1	Load immediate address
LA	1	Load absolute
FLA	2	Load absolute floating point
LBD	1	Load byte direct
LBI	1	Load byte indirect
Privileged Area Content Control		
LPC	1	Load program control
LB	1	Load base
LBR	1	Load register to base
LBB	1	Load base to base
LPi	1 *	Load privileged (1, 2, or 3)

Designation	Index	Name
STPi	1*	Store privileged (1, 2, or 3)
LRPi	5*	Load registers and privileged (1, 2, 3)
SRPi	5*	Store registers and privileged (1, 2, 3)
LPTR	1*	Load privileged to register
LRTP	1*	Load register to privileged
EPR	1*	Exchange privileged/registers
LEB	1*	Load executive and branch
Arithmetic		
A	1	Add
AD	4	Add double
ALH	5	Add left half
ARH	5	Add right half
AR	1	Add registers
AA	1	Add magnitude
AAD	4	Add double magnitude
FAR	2	Floating point add registers
FA	2	Floating point add
FAD	3	Floating point add double
FAA	2	Floating point add magnitude
FAAD	3	Floating point add magnitude double
S	1	Subtract
SD	4	Subtract double
SLH	5	Subtract left half
SRH	5	Subtract right half
SR	1	Subtract registers

Designation	Index	Name
FSR	2	Floating point subtract registers
FS	2	Floating point subtract
FSD	3	Floating point subtract double
M	1	Multiply
MLH	5	Multiply left half
MRH	5	Multiply right half
MR	1	Multiply registers
FMR	2	Floating point multiply registers
FM	2	Floating point multiply
FMD	3	Floating point multiply double
D	1	Divide
DLH	5	Divide left half
DRH	5	Divide right half
DR	1	Divide registers
FDR	2	Floating point divide registers
FD	2	Floating point divide
FDD	3	Floating point divide double
Formating Instructions		
STF	3	Scaled to floating point
STFD	4	Scaled to floating point double
FTS	3	Floating point to scaled
FTSD	4	Floating point to scaled double
DTF	3	Decimal to floating point
DTFD	4	Decimal to floating point double
FTD	3	Floating point to decimal
FTDD	4	Floating point to decimal double

Designation	Index	Name
Bit Manipulation		
SLL	1	Shift left logical
SLLD	1	Shift left logical double
SRL	1	Shift right logical
SRLD	1	Shift right logical double
SLA	1	Shift left arithmetic
SLAD	1	Shift left arithmetic double
SRA	1	Shift right arithmetic
SRAD	1	Shift right arithmetic double
SLC	2	Shift left and count
SLCD	4	Shift left and count double
RLL	5	Rotate left logical
RRL	5	Rotate right logical
LAN	1	Load and logical
LIR	1	Load inclusive or
LER	1	Load exclusive or
STAN	1	Store and logical
STIR	1	Store inclusive or
STER	1	Store exclusive or
LRAN	1	Load register and logical
LRIR	1	Load register inclusive or
LRER	1	Load register exclusive or

Designation	Index	Name
Branching		
C	1	Compare
CD	4	Compare double
FC	2	Floating point compare
FCD	3	Floating point double compare
TIS	1	Test indicators set
BZ	1	Branch on zero
BNZ	5	Branch on nonzero
BP	1	Branch positive
BNP	5	Branch not positive
BN	1	Branch negative
BNN	5	Branch not negative
DCBP	1	Decrement count branch positive
ICBL	1	Increment count branch low
DRBP	1	Decrement register branch positive
IRBL	1	Increment register branch low
TBAN	1	Test bits logical and
TBER	1	Test bits exclusive or
B	1	Branch
BLR	1	Branch linked registers
BLRE	1	Branch linked registers to executive

Special Purpose

A. Input/Output

Designation	Index	Name
BIO	1*	Begin input/output
HIO	1*	Halt input/output
TIO	1*	Test input/output

B. Multiprocessor

Designation	Index	Name
LID	1*	Load identity
TSS	1	Test/set and skip
DST	1	Delay and store time

C. Arithmetic and Programmed Arithmetic

Designation	Index	Name
PS	1	Programmed sine
PFS	2	Programmed floating point sine
PC	1	Programmed cosine
PFC	2	Programmed floating point cosine
PE	1	Programmed polynominal expansion
PFPE	2	Programmed floating point polynominal expansion
PAT	1	Programmed arctangent
PFAT	2	Programmed floating point arctangent
PSQ	1	Programmed square root
PSQD	4	Programmed square root double
PFSQ	2	Programmed floating point square root
PFSQD	3	Programmed floating point square root double

D. Repetitive Operations

Designation	Index	Name
DCRN	5	Decrement count and repeat next
VDP3	1	Vector dot product (dimension 3)
FVDP3	2	Floating point vector dot product (3)

Description	Index	Name
VCP3	1	Vector cross product (3)
FVCP3	2	Floating point vector cross-product (3)
FMMV3	2	Floating point matrix X vector (3)
LCMB	1	Limit check multiple bytes
IBRN	5	Increment base and repeat next
ILC	1	Increment and limit check
IBLC	5	Increment and byte limit check
IFLC	2	Increment and floating point limit check
ITLA	1	Increment test logical and
ITER	1	Increment test exclusive or
IFDP	2	Increment and floating point dot product
IBDP	5	Increment and byte dot product
IMDP	2	Increment and mixed dot product
ISB	5	Increment and sum bytes
E. Configuration Control		
LCCU	1*	Load configuration control unit
LCCE	1*	Load configuration control external
SCCU	1*	Store configuration control unit
SCCE	1*	Store configuration control external

2.3.7.2 Examples of Usage

The following paragraph presents examples in different areas of Space Station onboard digital data processing. The area and specific processing functions are identified, and the key instructions for performance are presented. Instruction mnemonic (operation) codes are as previously defined. The naming conventions are for the convenience in associating the definition of the instruction (SUMC operation) with the programming language. The translation of the mnemonic and the following program generation (assembler) features are assumed.

A. Immediate data formating capability

B. Symbolic definitions of absolute values.

C. Default base register assignment.

Executive Processing

A significant number of operations within the Executive involve maintenance of data in scratch-pad memory at any time a program task switch occurs (Figure 2-37).

A. Entry and Exit to the Executive

1. Function:

Old status → Executive storage → New status

2. Instructions:

a. Entry from current program (nonprivileged)

BLRE Stores Program Counter, BO, and Status

Note: The three corresponding items of the Executive are loaded.

EXECUTIVE PROGRAM

(Any series of Executive instructions to store status in main memory, process request, and select next program for execution.)

LEB Z Restores Program Counter, BO, and Status.

b. Exit is made to next instruction following the above BLRE on any referenced location (Z).

B. Redefining Scratch Pad Memory areas

1. Function:

a. Active GPR's → Privileged Area 1

b. Main Memory → Active GPR's → Area 1

2. Instructions:

(Assume B0 established on switch to executive and current status (B0, program counter, and program status information) is stored in the executive at the new B0 +) (location 3 of Area 1).

EPR 4, 1 (SPM Executive Data → $GPR_{4, 5}$)

Determine reason for switch to the Executive and the need for a complete switch of GPR.

a. Assuming Area 3 is available and B0 is still established.

```
    LRP3      STACK, 5    (new registers → GPR → Area#3)
```

b. Assuming Area 3 is not available then:

```
    STM       STACK, 4    (GPR → inactive stack)
    LM        STACK, 5    (ready stack → GPR's)
```

Instruction Sequencing

The following instructions are included to demonstrate two types of instruction sequence control.

A. Control of Program Instruction Loops

1. Functions:

Repeat the execution of a set of instructions N times, generating an index register value from (a) N-1 to 0 and (b) 0 to K (N-1) by K.

2. Instructions

a.
```
LI     RI, N-1     (assembler generated value to RI)
LOOP A (any set of valid instructions, RI may be modified.)
  DCBP     RI, LOOP A     (cycle if RI-1 ≠ negative)
```

b.
```
LI     RI, K(N-1), R0     (Limit to R1, reset R0)
LI     RK, K
LOOP B (again any valid instructions)
   IRBP     R0,     LOOP B, RK (cycle if RI ≠ negative)
```

B. Control of Abnormal/Infrequent Conditions

1. Functions:

Detect any overflow, underflow, divide check, or negative square root attempt in a series of calculations:

2. Instructions:

```
LPC     TRAP     (detect any exception flags)
(arithmetic operations)     (Program control can force branch)
B      END     (or any normal exit)
TRAP     (exception processing)
```

NOTE: Loading R and changing the LPC argument to R, TRAP will accomplish selective condition monitoring.

Specialized Processing of Standard Words

Many GNC equations containing the following processing functions frequently occur in control equations and each set of equations are repeated several times per second.

A. Vector Dot Product (Dimension 3)

1. Function:

$$M \ \ N = M_oN_o + M_1N_1 + M_2N_2$$

2. Instruction: (fixed-point)

```
LB      B2, M     (address reference → B2)
VDP3    R2, N     (M∘M → R2, R3)
```

B. Vector Cross Product (Dimension 3)

1. Function:

$$C = \begin{bmatrix} C_0 \\ C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} M_1N_2 \\ M_2N_0 \\ M_0N_1 \end{bmatrix} \begin{matrix} - & M_2N_1 \\ - & M_0N_2 \\ - & M_1N_0 \end{matrix}$$

2. Instruction: (floating point)

```
LB       B2, M     (address reference → B2)
LB       B3, N     (address reference → B3)
FVCP3    C         (M x N → C, C + 1, C + 2)
```

Special Processing of Partial Words

Experiment data processing frequently involves operations on sets of data where each element is an encoded 8-bit value or the standard 32-bit SUMC words. Statistical operations such as determining standard deviation or correlation coefficients and numeric techniques for curve fitting or analysis involve repeated application for the following operations:

A. Summation of a Vector or of 8-Bit Values

1. Function:

$$\text{Weight} = \sum_{0}^{N-1} Y_i$$

2. Instruction:

```
a.                                (assuming self-incrementing
                                  implementation)
      LB        B2, Y             (address reference → B2)
      ISP       R2, N/4-1         (sum N bytes)
b.                                (assuming commanded repetition
                                  implementation)
      IBRN      R3, N/4-1         (establish loop value)
      ISP       R2, Y, R3         (sum 4 Bytes of Y)
```

B. Dot Product of Two Eight-Bit Vectors

1. Function:

$$M \; Y = \sum_{i=0}^{N-1} M_i Y_i$$

2. Implementation:

```
a.                                (assuming self-incrementing
                                  repetition)
      LB        B2, M             (address reference to B2)
      LB        B3, Y             (address reference to B3)
      IBDP      R2, N/4-1         (accumulate products)
b.                                (assuming commanded repetition
                                  implementation)
      LB        B2, M             (address reference to B2)
      IBRN      R3, N/4-1         (establish loop control)
      IBDP      R2, Y, R3         (Dot product 4 bytes of Y)
```

2.3.7.3 References

1. MSFC Advanced Aerospace Computer, Sperry Rand Corporation, Report No. SP-232-0384, Contract No. NAS8-20055, 6 July 1970.

2. Proposed Instruction Set for the SUMC System, Sperry Rand Corporation, Report No. SP-232-040501, Contract No. NAS8-20055, 4 September 1970.

3. An Application-Oriented Multiprocessing System. IBM Corporation, IBM Systems Journal, Volume 6, No. 2, 1967.

4. Data Bus Terminal Buffer Selection Rationale, Newsletter No. IBM/SPE-96, 21 June 1971.

2.4 EXECUTIVE PROGRAM

A preliminary design specification for the SUMC Executive program has been prepared in CPCEI format. The specification is provided as Appendix A to this document.

PRECEDING PAGE BLANK NOT FILMED

Section 3
DATA BUS

3.1 REQUIREMENTS

The specific data distribution system requirements depend on the quantity and type of data generated, the maximum allowable response times associated with that data, and the number of stations or terminals gathering data.

Data quantities and rates were assessed for both experiments and subsystems.

The subsystem analysis was based upon a tabulation of Space Station subsystem requirements relative to the data bus. These requirements were derived from information provided by the subsystem designers which included a description of each data item, specifying the data source, category (stimulus or response), form (analog, bilevel, etc.), usage, and occurrence statistics.

Load sheet data was keypunched and input to a computer program which performed various sorts and tabulations of the data and provided the results in printed report form. Results are based upon a total of 9,580 data items and include all subsystem requirements.

The following is a summary of data rates by subsystem. These rates represent actual data, and are average (not peak) rates. Address bits and other overhead is not included, nor is the effect of preprocessing such as remote limit checking.

Subsystem	Data Rate
GNC	3.60 kbps
EC/LS	3.06 kbps
Communications	32.16 kbps
Propulsion	0.90 kbps
Data Management	134.81 kbps
Electrical Power	7.96 kbps
Total	182.49 kbps

An analysis of data block lengths revealed that approximately 99 percent of the data items are two bytes (sixteen bits) or less in length, not including overhead. These items consist primarily of single-parameter measurements and commands. It is significant to note, however, that these items make up only about 10 percent of the data volume. The other 90 percent consists of data to and from the communications subsystem, displays, and bulk digital storage devices, and other multiword transfers. These transfers are generally of longer block length, ranging from 64 bytes upward to over 1,000 bytes. Examples of these are listed below:

RDAU Block Dump (64 channels at 8 bits each)*	64 bytes
RDAU Memory Load* (64 channels at 20 bits each)	160 bytes
Display Update (1,000 characters at 8 bits each)	1,000 bytes
Data to/from Communications (Shuttle, DRS, etc.)	Indeterminate

*Based upon 64 channels per RDAU. Baseline is now 32 channels.

These results suggest that the majority of routine bus transactions may be accommodated by data words of 16 bits or less, but that special arrangements may be required to effectively handle a relatively few data sources characterized by high data volume and long message lengths.

The subsystems were then analyzed to determine the requirements for direct terminal-to-terminal data transfers, bypassing the DMS data processor. Requirements for transfers of this type were found to be minimal, since the majority of data does require processing of some type. This conclusion is based upon the Space Station DMS baseline which centralizes the data processing functions, including those associated with GNC. Thus, the bulk of subsystem data, such as commands, measurements, and display information either originates in the DMS computer or must receive processing such as scaling, code conversion, and formating.

Sorts were also made on the data requirements listings to determine the incidence of randomly occurring versus cyclic or periodic data. The results are as follows:

Periodic Data	32.30 kbps (17%)
Random Data	150.19 kbps (83%)

The above are average data rates and do not include overhead or preprocessing such as RDAU limit checking. The periodic portion consists primarily of status monitoring and display data, while the random portion consists of command and control functions, checkout and fault isolation data, communications, and memory transfers.

An analysis was then performed to determine the reduction in data bus traffic which could be achieved by performing limit checking in the RDAUs. This technique is applicable to those functions requiring periodic status testing which would otherwise be performed in the DMS computer.

As stated previously, the data bus load contributed by periodic operations was calculated as being 32.30 kbps. Of this amount, 50 percent or 16.14 kbps is associated with periodic status monitoring (i.e. limit checking) of approximately 3,100 parameters. This function could be performed at the RDAU level. Under this approach the DMS computer would still be required to sample individual parameters at periodic but relatively infrequent intervals to verify proper operation of the local limit check circuitry, and would be required to load and update the local RDAU memories. The additional bus load associated with these operations has been calculated as being approximately 1 kbps, assuming one computer sample of each limit check parameter each 5 minutes and updating of the RDAU memories once each hour. The net reduction in data bus traffic is therefore on the order of 15.0 kbps, or 8.2 percent.

Finally, a sort of the data listing was made to determine the bit rate associated with command and control functions. The average command bit rate was found to be approximately 600 bits/second. The primary contributor to this total is the GNC subsystem at 560 bits/second, with the balance being scattered among the other subsystems.

The amount of experiment data is derived from the experiment descriptions in the Blue Book. The experiment data totals were 2.4 MBPS and 7.2 MBPS for ISS and GSS, respectively. The rates shown involve two assumptions:

- High rate experiments can be scheduled such that only one is operating at any given time.
- The maximum data rate to be handled by the data bus will be such as to incorporate only the most common occurrences.

GSS Experiments A-33CC, ATM, and A-4A, Narrow Field UV Telescope involve 9.5 MBPS and 8.6 MBPS associated data rates, respectively.

These will be handled by special data handling methods as will the 50 MBPS generated from the ES-1AA, Earth Observation. However, there are three experiments which generate 7.0 MBPS, A-5A, X-Ray Telescope, A-5B, Gamma Ray Telescope, and A-6, IR Telescope. Consequently, the data bus should be capable of handling at least 7.2 MBPS peak data rate. This rate corresponds to a total data bus rate of approximately 10 MBPS when overhead and control bits are added.

The data distribution system must also allow a high rate experiment package to operate at the same time other high rate transfers occur, e.g. 10 MBPS data transmission over the communication system to the ground. Thus, more than one 10 MBPS bus is required.

The analog requirements imposed on the data distribution system are derived from the baseline requirements documents. These requirements have not been modified extensively from those requirements derived for the 33 ft diameter Space Station.

3.2 ALTERNATE CONCEPTS AND TRADE STUDIES— SPACE STATION

3.2.1 Data Bus Control

Data bus access must be controlled to eliminate scrambled data or erroneous commands from simultaneous transmissions. Three methods of controlling a digital data bus are:

A. Polling

B. Interrupt

C. Contention

3.2.1.1 Polling Approach

Under control of a data bus supervisor, all input and output devices are sampled at a rate greater than the minimum required to satisfy system dynamics and operation. Units on line respond only on request from the controller. Most polling schemes do not require each device to be interrogated at equal rates but at a frequency which allows system response time requirements to be met regardless of other input/output device specifications.

For control unit to remote terminal data transfer, a control word is put on the bus following synchronization. Data to the terminal follows. All remote terminals decode the control word and the proper terminal accepts the data. For data transfer from a remote terminal to a Central Processing Unit (CPU), a command is sent by the control unit. All terminals decode the control word and the proper terminal responds.

Hub polling is a slight modification to the standard polling technique. Terminals on the data bus are sequentially addressed. If a terminal has no data, it sends the polling signal to the next terminal. In the event the terminal has data to transmit, the polling signal is not transferred until the end of the data transmission. This method reduces propagation delays but results in a lack of flexibility since the entire polling loop must be completed before the bus control unit regains control.

The polling approach results in simple input/output device interfaces and can be either hardware or software controlled. Software complexity can be reduced by using stored program sampling formats. Hardware polling can be used where sampling priorities allow fixed polling sequences.

3.2.1.2 Interrupt Approach

This design concept allows a remote terminal to acquire the bus controller when data is ready or when attention is required. The controller then addresses the terminal as in the polling scheme. In effect, the polling and interrupt approaches can be integrated with the interrupt function used to modify the polling sequence.

The interrupt concept requires another bus, since hardwired interrupts from each remote terminal is undesirable. However, data bus transfer rates and software complexity can be reduced by incorporating a simple interrupt scheme.

3.2.1.3 Contention Approach

In a contention system, each terminal competes for the use of the bus. A "line available" signal or polling signal, circulates through each terminal when the bus is free. The terminal acquires access to the bus by capturing the polling signal. The terminal may then transfer data.

The bus controller controls the "line available" signal. When a command is to be sent to a specific device, the controller removes the polling signal from the line and inserts the necessary command. Consequently, this approach can also be used in conjunction with the polling scheme described previously. Since the bus controller holds the "line available" signal during a command or a polling sequence, provision must be made for the terminal to respond on command without the "line available" signal. This system allows immediate response by any commanded device since other devices do not have the capability to capture the line. It also provides a means of acquiring data and controlling terminals not normally involved with the contention system.

Successful operation in a contention system depends on low-data bus load factors and the assurance that no terminal will capture the line for long periods. This requirement may not allow computer-to-computer or bulk-storage transfers.

Advantages of the contention mode are minimized bus data rates and simplified software. However, terminal hardware has to be more complex to provide the intelligence for capturing the polling signal and outputting data at proper times.

3.2.1.4 Command/Data Channel Commonality

The digital channels are operated in a half duplex mode. The amount and frequency of commands and/or instructions are relatively small compared to the amount of data. Buffers at each remote terminal allow data to be stored when urgent commands are to be transmitted. Thus, no data loss occurs even when data transmission is interrupted for priority commands. (Software control of the polling sequence provides for interrupts of this nature.) Incorporating commands on the same channel as data allows a more efficient design to be effected. As an example, assume two separate 10 MBPS digital buses. On the surface, it appears that this represents a total capacity of 20 MBPS. However, if one line is used solely for commands having a rate of 2 MBPS, the total data transfer is only 10 MBPS. Consequently, incorporating a separate command line either involves more design effort and less hardware commonality for two lines having different speeds or less efficient use of total system capability.

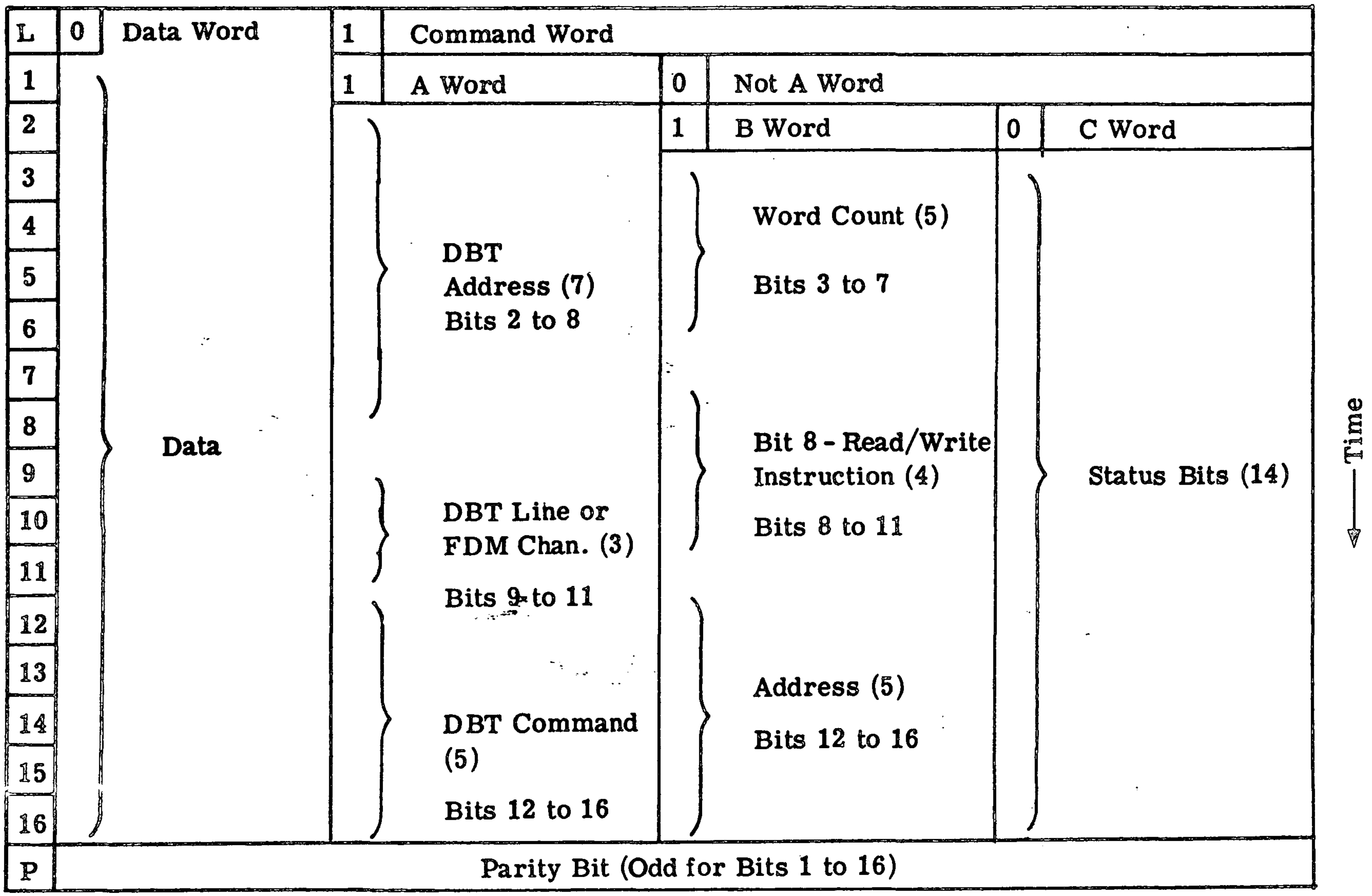
3.2.1.5 Digital Data Bus Operational Control

All communications or messages in data bus digital channels will either be "commands" generated by the controller or "responses" generated by one of the Data Bus Terminals (DBT). These messages will be transmitted in an ordered sequence and consist of a combination of 18 bit words.

Figure 3-1 shows the word formats for information transfer over the data bus. Four types of 18 bit words are identified as "A", "B", "C", or "D" words. An "A" word contains the DBT address and commands; a "B" word contains a word count, the device instruction, and the I/O channel address; a "C" word contains status; a "D" word contains data. Command messages occur in the form of an "A" word, followed by data if any, and ending with a "B" word. A normal response to a command consists of the echoed "A" word, followed by up to 32 data ("D") words, and ending with a "C" word. Each bus transmission starts with a synchronization burst consisting of 15 zeroes followed by a three-bit sequence of one, zero, one. This is followed by an "A" word, data if any, and ends with either a "B" or "C" word. In a device-to-device transfer, an additional "A" and "B" word (designated hereafter as "$\bar{A}'$" and "$\bar{B}'$") are transmitted from the controller to the DBT as two data words, i.e., having identical formats as an "A" and "B" word but containing a lead zero rather than a lead one. This data provides the address of the second device and contains any necessary control information. The DBT then reinserts the lead one's and uses these data words as control words for device-to-device transmissions.

The data bus controller receives information containing the status of the DBTs and RDAUs and or experiments associated with a particular DBT from the status of the "C" word bits. The arrangement of the "C" word is shown in Figure 3-2.

Transfers between the controller and DBT are illustrated in Figures 3-3 and 3-4. Figure 3-3 shows the normal sequence for transfer of data from the DBT to the controller (identified as a read mode). In a typical read sequence, the controller transmits synch, an "A" word and a "B" word to the DBT. In response, the DBT generates a synch burst, an echo "A" word, up to 32 16-data bit plus two control bit data words, and ends with a "C" word containing terminal status information. The total response time (from time



L = Lead Bit
P = Parity Bit

Figure 3-1. Data Bus Word Formats

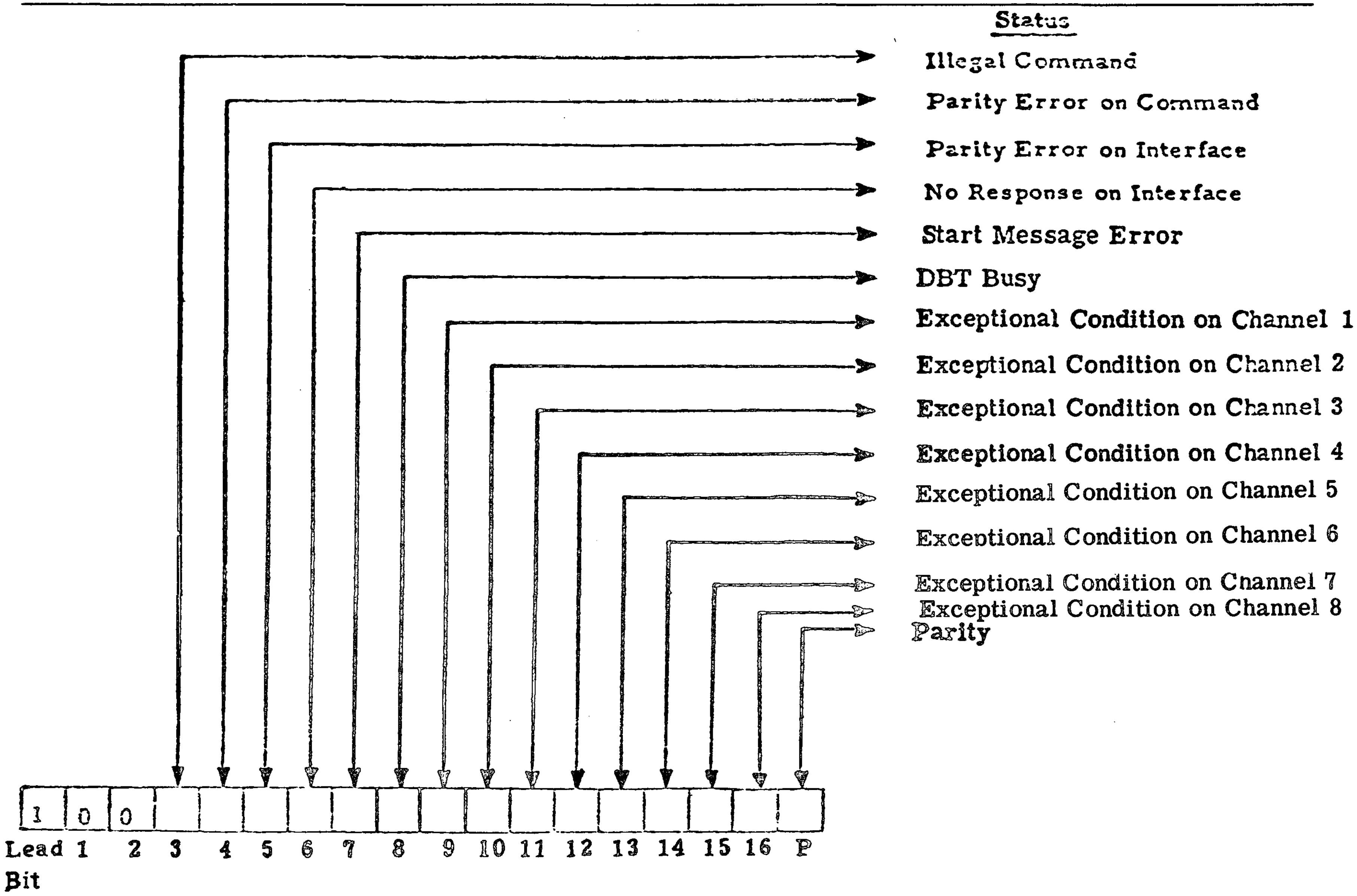

Status
Illegal Command
Parity Error on Command
Parity Error on Interface
No Response on Interface
Start Message Error
DBT Busy
Exceptional Condition on Channel 1
Exceptional Condition on Channel 2
Exceptional Condition on Channel 3
Exceptional Condition on Channel 4
Exceptional Condition on Channel 5
Exceptional Condition on Channel 6
Exceptional Condition on Channel 7
Exceptional Condition on Channel 8
Parity
1
0
0
Lead Bit
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
P

Figure 3-2. C-Word Format

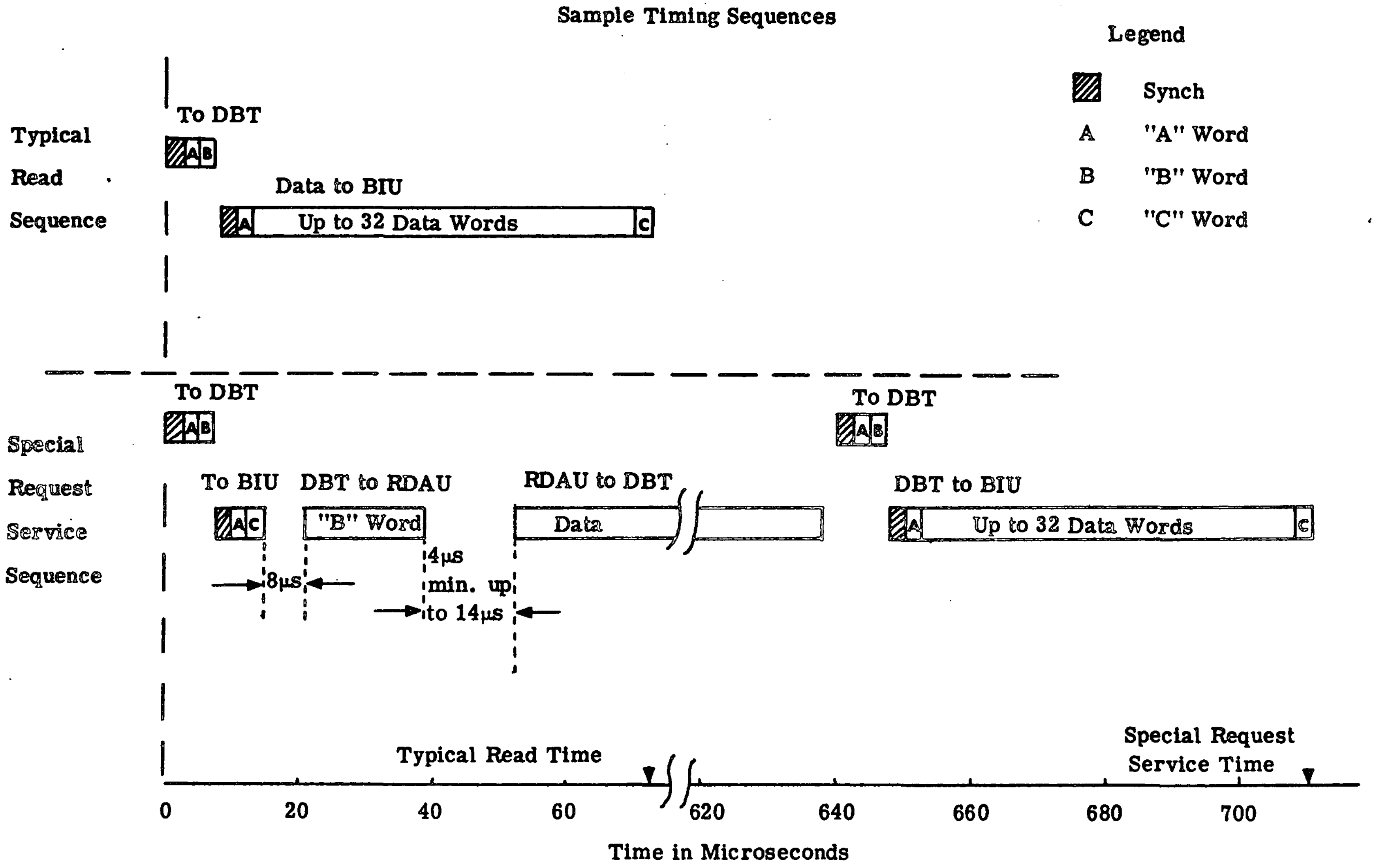


Figure 3-3. Read Sequence Timing

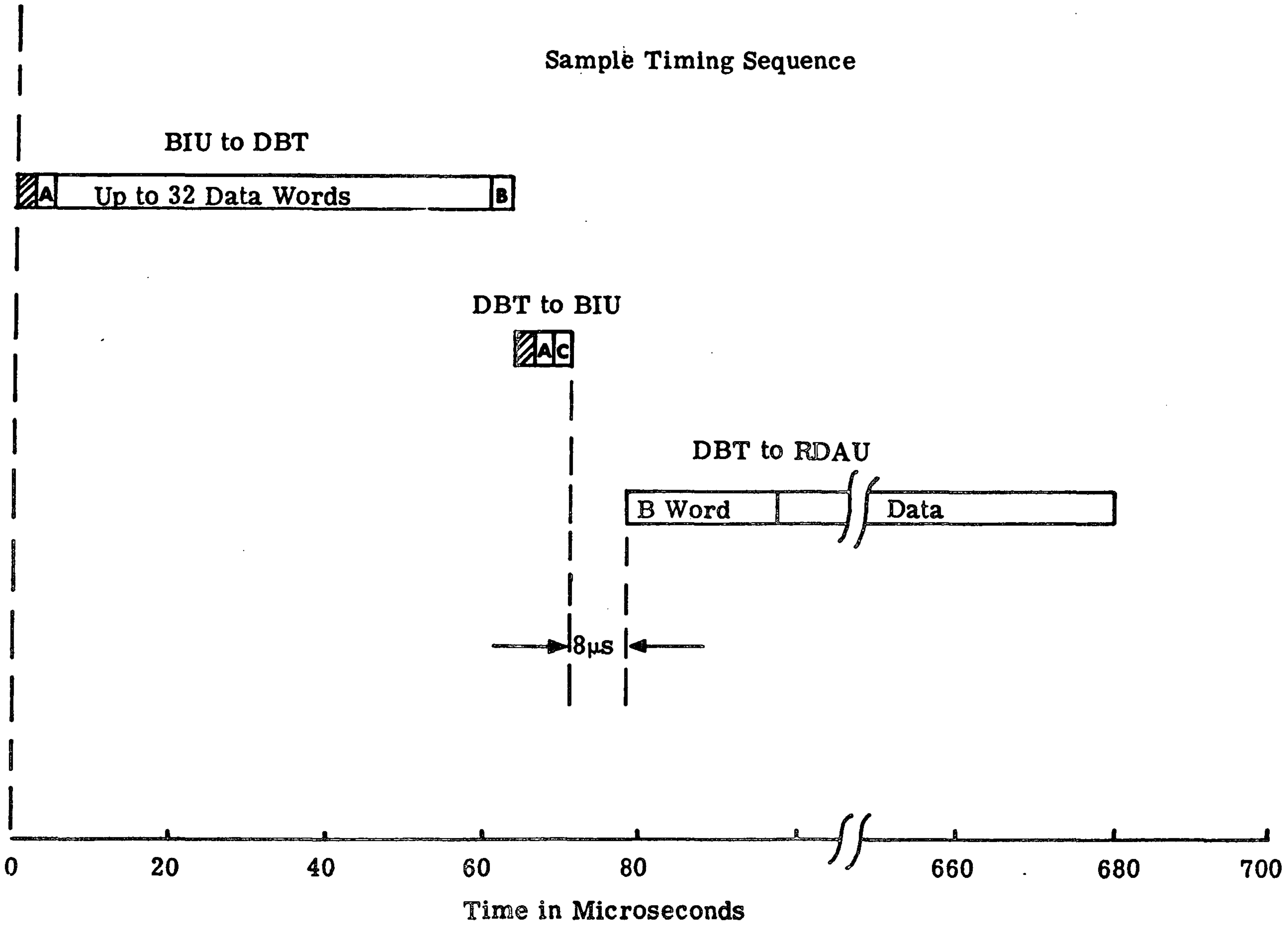


Figure 3-4. Write Sequence Timing

of command initiation to receipt of message) is approximately 73 microseconds with circuit and propagation delays accounting for approximately 4 microseconds. In the event special data is required from an I/O device such as a Remote Data Acquisition Unit (RADU), a special request sequence must be serviced whereby a large time delay of 638 microseconds is allowed for data transfer from an I/O device to the DBT at 1 megabit per second. The Controller issues a command consisting of synch, an "A" word, and a "B" word. The DBT responds with a synch, an ehoed "A" word, and a "C" word. After a delay of 8 microseconds, the DBT transmits the "B" word to the I/O device (hereafter assumed to be an RDAU). The RDAU responds within 4 to 14 microseconds with the data transfer taking up to 576 microseconds for a total elapsed time of approximately 638 microseconds allowing for circuit and propagation delays. The typical read sequence is then initiated to bring data to the BIU. Total elapsed time will be less than 710 microseconds. The data bus channel is only occupied for 85 microseconds and other functions can be performed during the channel free time.

Figure 3-4 illustrates the typical write mode. The controller transmits a synch, an "A" word, up to 32 18-bit data words, and a "B" word, requiring 63 microseconds. The DBT responds with a synch, an echoed "A" word, and a "C" word. After a delay of 8 microseconds, a "B" word is transmitted to the RDAU followed by up to 32 data words. Total elapsed time will be less than 638 microseconds. The data bus channel is only occupied for approximately 12 microseconds however, and can be utilized to perform other functions.

The sequence for terminal to terminal information transfer is shown in Figure 3-5. In the event both terminals are not on the same channel, a preliminary command must be sent to switch one of the terminals (Identified by the address of the "A" word) to the other's frequency channel. This sequence is not shown in the figure. The controller transmits a synch, an "A" word, an "$\bar{A}'$" data word, a "$\bar{B}'$" data word, and a "B" word. The DBT responds with a synch, an "A" word, and a "C" word. After a nominal delay time (approximately 1 microsecond), the DBT transmits another synch, an "$\bar{A}'$" word, up to 32 18-bit data words, and a "$\bar{B}'$" word. The DBT identified

4

R266

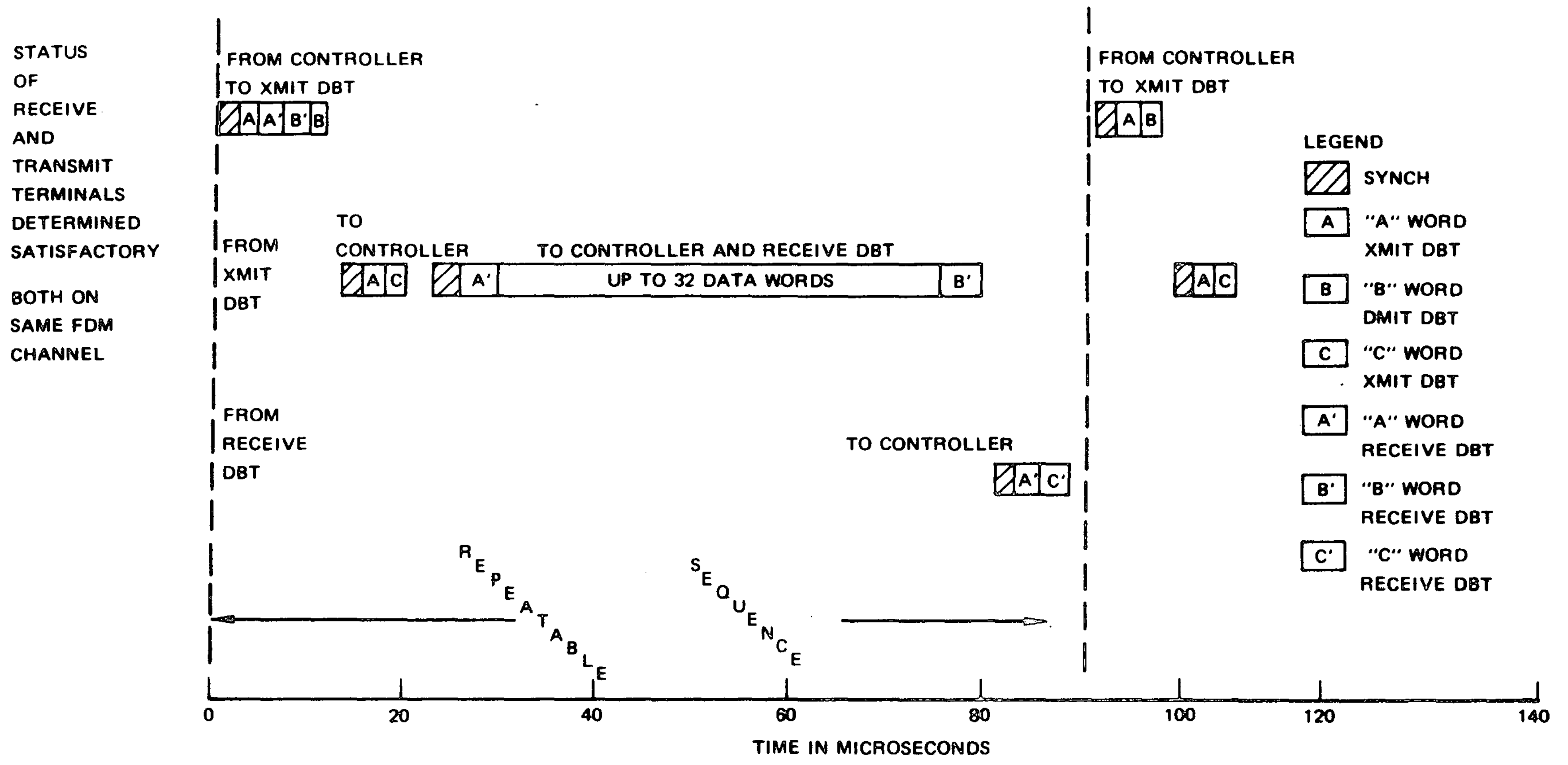


Figure 3–5. Terminal–to–Terminal Transfer (TTT) Timing Sequence

by the address field in the "$\bar{A}'$" word receives the data and responds with a synch, an echoed "$\bar{A}'$" word, and a "$\bar{C}$" word. This sequence is repeatable with "$\bar{A}'$" and "$\bar{B}'$" words issued with each terminal to terminal transfer sequence. The total sequence will require less than 84 microseconds but occupies the data bus channel for the full amount of time. Therefore, the data bus channel is occupied for the total length of time required for terminal-to-terminal transfer. The reset command is issued to the DBT identified by the "A" word to reset the FDM channel, if necessary, and to reset the registers containing the "$\bar{A}'$" and "$\bar{B}'$" words.

3.2.1.6 Data Bus Controller

A trade study has been performed to determine the applicability of using a specially designed data bus controller versus using executive routines of the main processor for implementing bus control. The proposed capabilities for the SUMC processor were used as a base for the software approach to data bus control.

The specially designed data bus controller would operate in conjunction with the SUMC processor but would handle all the periodic data request functions. This controller would be interruptable by the processor in the event special data was needed. The trade study results are shown in Table 3-1. This study indicates that a separate data bus controller adds an additional black box to the system, increases hardware costs by 19 percent and saves very little software. Therefore, data bus control will be designed into the SUMC processor executive routines.

3.2.1.7 Error Detection Techniques

Data bus control, to be complete, must have the capability of detecting when errors occur. The Space Station data bus system will employ high signal-to-noise ratios in any given channel. Consequently, if a binary symmetric channel is assumed for the digital data channels, the probability of any bit being in error will be less than 10^{-6}.

Although this magnitude of error rate is not high, message transmissions require greater assurance that they are accurately received. Thus error

Table 3-1
TRADE SUMMARY DATA BUS CONTROLLER

	Computer Integrated/ Dedicated Controller	Free-Running Controller	Remarks
Hardware:			
Control function cost (less memory) (dollars)	270, 000	320, 000	Free running approach requires 20 percent additional hardware.
Software:			
SUMC degradation	0.003 percent	---	Deltas are negligible.
Control function cost (less commands) (dollars)	194, 000	208, 000	
Programmatic:	679 points	610 points	

control techniques are necessary. In general, error control implies the detection and correction of errors. There are two methods of implementing the correction process after an error is detected. These methods are by retransmission or by forward error correction via coding. The later technique is not advantageous unless no return path is available to request a retransmission or unless the uncoded error rate is very high. In addition, forward error correction requires relatively complex hardware. Since the Space Station data bus involves low error rates and employs half duplex operation (i. e., a return path is available), when error correction is required retransmission is the recommended approach.

Fundamental to any error control is the statistics of the perturbing noise. In most communications below 10 MHz, man-made burst noise generally predominates. This is partially because low frequency, externally generated noise is difficult to attenuate via shielding. Cable shields, however, become effective above 10 MHz and are especially effective above 40 MHz. When man-made noise is not present, as approximated by a single channel data transmission operating over coaxial cables above 100 MHz, Gaussian noise statistics provide an accurate approximation. Since the Space Station data bus encompasses a wide frequency spectrum, both Gaussian and burst noise statistics are of interest.

The detection of errors provides various advantages and is mandatory to complete the retransmission/correction process. The following techniques are considered as candidates for error detection:

A. Simple parity
B. Dual parity
C. Dual message transmission
D. Cycle block coding (double error detection)
E. Data feedback
F. Waveform validity checking

The following assumptions were utilized in the analysis of the various schemes of error detection:

A. A binary symmetric channel is assumed. This implies that no a priori information is available and Gaussian noise statistics apply. An error in detectinga one when a zero is transmitted has the same probability as an error in detecting a zero when a one is transmitted.

B. Unless otherwise indicated, all data is in NRZL format.

C. Bit error rates are less than 10^{-5}. The block length times the bit error rate is much less than 1.

D. All transmission is serial and simplex.

E. Transmission to and from each station has the same error probability.

F. Burst error detection is noted but not formulated. A burst error is defined as erroneous bits that result in a noise induced signal level of one polarity (binary zero or binary one) for b consecutive bits.

The different types of error detection schemes were evaluated for the relative error probability, undetected error probability and through-put efficiency. Each of the error detection techniques are discussed in the following paragraphs:

Simple Parity

Simple parity is the case where a single check bit is added to (n-1) data bits such that the sum of the n bits is odd (or even in case of even parity).

For simple parity $P_b = np$ approximates the probability of an error in a group of n bits. Simple parity will detect all odd errors. The error probability is obtained by calculating the probability of occurrence for all even errors. However, an approximate probability can be calculated by

$$P_u = E_2 = \frac{n!}{2!\,(n-2)!}\,P^2 = \frac{n(n-1)}{2}\,P^2$$

The above equation considers all multiple errors and is conservative in that simple parity actually detects all odd multiple errors. This is typical in that most coding techniques will detect many error combinations but only the most probable are considered. The approximations, however, are accurate for binary symmetric channel since the probability of three errors is at least an order of magnitude less than the probability of two errors.

Through put efficiency is (n-1)/n, since in a block of n bits, n-1 are data.

Dual Parity

Dual parity is here by defined by dividing a block of n bits into even and odd groups and assigning a simple parity to each. Thus any two consecutive errors will be detected.

The error probability is as for simple parity $P_b = np$.

The undetected error probability can be computed as for single parity with n/2 substituted for n.

$$P_u = \frac{\left(\frac{n}{2}\right)\left(\frac{n}{2}-1\right)}{2} P^2 = \frac{1}{4}(n)\left(\frac{n}{2}-1\right) P^2$$

Throughput efficiency is (n-2)/n.

Dual Transmission

This technique is typified by the transmission of a message followed by the compliment of the same message. With dual transmission, twice the number of bits are required with each bit being dual redundant. Thus, the probability of an error is double that of no redundancy.

$$P_b = 2np$$

The probability of any data bit error being undetected is equal to p^2, the probability of two independent failures. Since there are n message bits,

$$Pu = np^2$$

Throughput efficiency, $n/2n = 1/2$

Data Feedback

This technique consists of sending a message to the receiver followed by the receiver retransmitting the same message back (before accepting it). A comparison is then made to verify accuracy and if no errors are detected, an accept signal is transmitted to cause acceptance. The probability of one or more errors is,

$$P_b = 2np + ap$$

Where a is the number of bits in the Accept/Reject signal.

The probability of an undetected error, Pu, is np^2. Throughput efficiency is less than 1/2 because of the accept signal. The key advantage of this technique is that a high confidence of proper equipment operation accompanies the transfer of a message. Burst error detection of n bits is also provided.

Block Cyclic Code (with double error detection)

Many cyclic codes are available and a paritcular code can be tailored to the specific requirements. Approximating the characteristics of most codes (including simple parity) tends to be conservative. Nevertheless, if we choose a general case of (n, k) where n is the block length and k is the number of data bits and assume that any two errors can be detected (no correction implemented), the probability of an error is np. The undetected error rate for three or more errors is approximated by the following equation:

$$E_3 = \frac{n(n-1)(n-2)(n-3)!}{3!\ (n-3)!} P^3 = \frac{n(n-1)(n-2)}{6} P^3$$

Throughput efficiency is k/n.

Waveform Validity of Manchester II Code

Data transmitted in formats other than NRZL can offer the advantage of error detection. This advantage is gained at the expense of higher frequency or more than two levels. Although it is not commonly recognized, Manchester II (Biphase L) has this property. This coding technique can be viewed as each bit being the consecutive transmission of two NRZL coded bits (in one bit period) with the second being an odd parity. With this observation, error probability approximations can be determined from equations for simple parity with a block length of 2. The error probability is 2np, twice that of simple parity although it should be noted that phase detection techniques can be used to reduce this to np.

Each bit's undetected error probability is determined from

$$Pu' = \frac{n'(n'-1)}{2} P^2$$

which for n = 2 is

$$Pu' = P^2$$

Since there are n bits in a block

$$Pu = np^2$$

The throughput efficiency is 1/2.

A burst noise pulse will tend to distort the waveform of a single bit such that a phase reversal (level change) does not take place. This constitutes an error which should be detectable. Thus, all burst errors can theoretically be detected if parity within each bit period is checked.

It should be noted that as the transmission rate increases, waveform validity checking becomes more difficult.

Block Parity

This technique, also referred to as vertical and horizontal parity, is often used in tape recording. A message is organized into c groups or characters of r bits each including parity check bits. The organization forms a block of r rows by c columns. The last bit in each column (or character) is a vertical parity check bit and the last column is horizontal parity check bits.

This form of checking will detect any three errors in the block and many combinations of multiple errors. It also detects all single burst errors of r bits and many multiple burst errors. In a binary symmetric channel, the error rate can be approximated by:

$$E_4 = \frac{rc(rc-1)(rc-2)(rc-3)}{4!} P^4$$

or simply

$$\frac{n(n-1)(n-2)(n-3)}{24} P^4$$

The throughput efficiency for this coding technique is:

$$\frac{(r-1)(c-1)}{rc}$$

where $n = rc$

Table 3-2 summarizes results and provides an example calculation for the tentative Space Station system with $p = 10^{-6}$ and $n = 16$. Combinations of the techniques indicated can greatly improve undetected error probabilities at the cost of throughput efficiency. However, for Space Station purposes, a probability of undetected error of 10^{-10} is considered adequate. Therefore, the use of simple parity is recommended for simplicity and high throughput efficiency.

3.2.2 Signal Design

The Space Station data bus employs an asynchronous time division and frequency division multiplex (TDM/FDM) scheme for digital communication

Table 3-2

ERROR DETECTION TECHNIQUE APPROXIMATION SUMMARY

						Example: $p = 10^{-6}$, $n = 16$ except as noted		
		P_b	P_u	T_R	Burst Error Detection	P_b	P_u	T_R
1.	Simple parity	np	$\frac{n(n-1)}{2}p^2$	$\frac{n-1}{n}$	Detects 50% of burst errors	1.6×10^{-5}	1.2×10^{-10}	94%
2.	Dual parity	np	$\frac{n\,\frac{n}{2}-1}{4}p^2$	$\frac{n-2}{n}$	Detects all single bursts of 2, 3, 5, 7 etc.	1.6×10^{-5}	2.8×10^{-11}	87%
3.	Dual transmission	2 np	np^2	$\frac{1}{2}$	Detects all single bursts of n and some longer	3.2×10^{-5}	1.6×10^{-11}	50%
4.	Dual feedback	$2\,np + \Delta p$	$>np^2$	$<\frac{1}{2}$	Detects all single bursts and verifies hardware	$>3.2 \times 10^{-5}$	$>1.6 \times 10^{-11}$	<50%
5.	Cyclic block code (n, k)—double error detection	np	$\frac{n(n-1)(n-2)}{6}p^3$	$\frac{k}{n}$	Detects burst error of 4 bits** codes can be chosen for better performance	1.6×10^{-5}	5.6×10^{-16}	73%
6.	Manchester II validity check	np or 2 np*	np^2	$\frac{1}{2}$	Detects all bursts	1.6×10^{-5} or 3.2×10^{-5}	1.6×10^{-11}	50%
7.	Block parity (c x r) n = cr	np	$\frac{n(n-1)(n-2)(n-3)}{24}p^4$	$\frac{(c-1)(R-1)}{cr}$	Detects all single bursts of r and many longer	(c=4, r=4) 1.6×10^{-5}	1.8×10^{-21}	56%***

P_b = Probability of one or more bits in error in block of n

P_u = Probability of an undetected error in a block of n

= Bit error probability

T_R = Throughput efficiency

*Depends on detection technique

**Based on Bose-Chaudhuri code with k-11, n=15, double error detection

***Longer blocks are more efficient but will have between data bus terminals and the main computer subsystem. Three major parameters have to be considered in the design of the data bus signalling techniques:

A. Modulation technique

B. Timing and synchronization

C. Waveform format

These parameters are interrelated and are discussed in the following paragraphs.

3.2.2.1 Modulation Techniques

Three modulation choices are amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM). With respect to digital transmission these are often called amplitude shift keying (ASK), frequency shift keying (FSK), and phase shift keying (PSK). As discussed in most communication texts, AM is a linear modulation process while FM and PM are nonlinear, expotential modulation processes which can easily be seen by describing the sinusoidal voltage E(t) in its complex form:

$$\begin{aligned} E(t) &= Ae^{j(\omega t + \theta)} \\ &= Ae^{j\omega t}e^{j\theta} \end{aligned}$$

The modulation variables for AM, FM, and PM are A, ω, and θ respectively. From the above, the similarities between FM and PM can easily be appreciated and in many instances FM and PM can be discussed interchangeably.

Four variations of AM are candidates for a frequency division multiplex system. These are double sideband (DS), double sideband suppressed carrier (DSSC), single sideband (SS) and vestigial sideband (VS).

Peak or average power is not a major restriction in the Space Station data bus design (although it is usually desirable to minimize power dissipation) except for limiting interchannel interference. Design of the bus will insure relatively high signal-to-noise energy ratios. With high signal-to-noise ratios, error rate performance for the various modulation schemes is comparable within a few db. Figure 3-6 compares the various modulation techniques and shows that for an error rate of 10^{-6}, required signal-to-noise ratios differ by about 4 db between the best and worst case. This means that the worst case (noncoherent FSK), can be equated to the best case (coherent PSK/Polar baseband) by employing 4 db of additional signal power. Figure 3-6 shows performance of various modulation techniques in a Gaussian noise channel, and similar relative performance has been shown to exist in the presence of burst noise.

In the following paragraphs, each of the potential candidate modulations schemes are discussed with emphasis on implementation. The more practical approaches are determined and a comparison table is presented.

Amplitude Modulation

Of the four variations of amplitude modulation, single sideband (SS) and double sideband suppressed carrier (DSSC) involve complex implementation, especially at the receiving end. The complexity results because the original carrier must be constructed before demodulation can take place. Reconstruction of the carrier is difficult, and improper reconstruction results in phase intercept distortion.

Of the various carrier modulation techniques, the simplest and least critical is generally accepted to be double sideband AM. The simplest of modulators is a keyed oscillator triggered on and off by a digital input, Figure 3-7. This technique, however, can produce considerable unnecessary sideband energy unless output filtering is employed. Output filtering requirements can be greatly minimized if standard linear modulation is employed and the input is prefiltered. These precautions do not preclude a simple design and greatly relax the requirements of output filtering. Complete reliance upon output filtering to avoid interchannel interference will make a fail-safe transmitter extremely difficult to achieve. Thus, a simple, yet effective approach to the AM modulator would appear to be that shown in Figure 3-7 with a standard balanced modulator. The balanced modulator approach

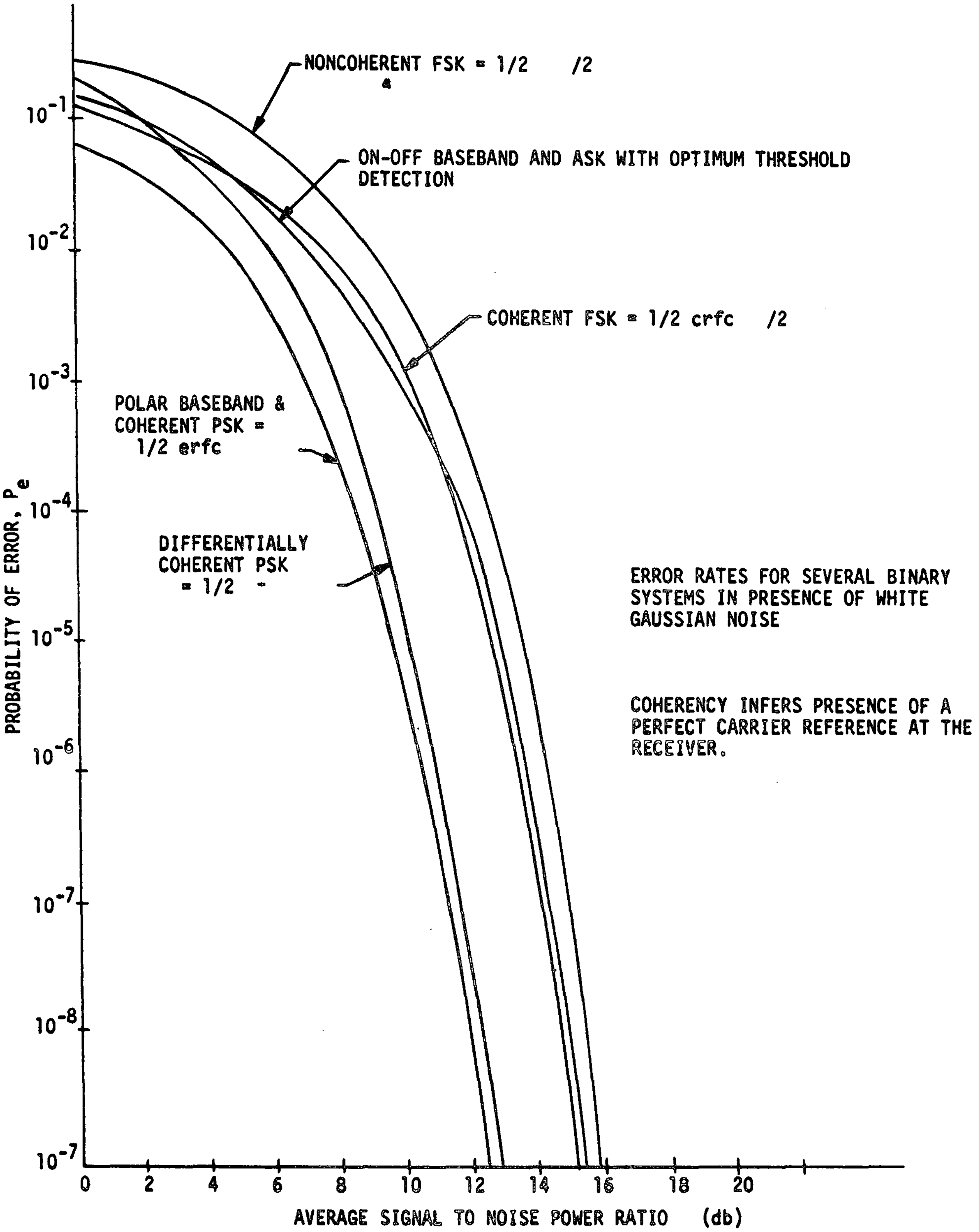
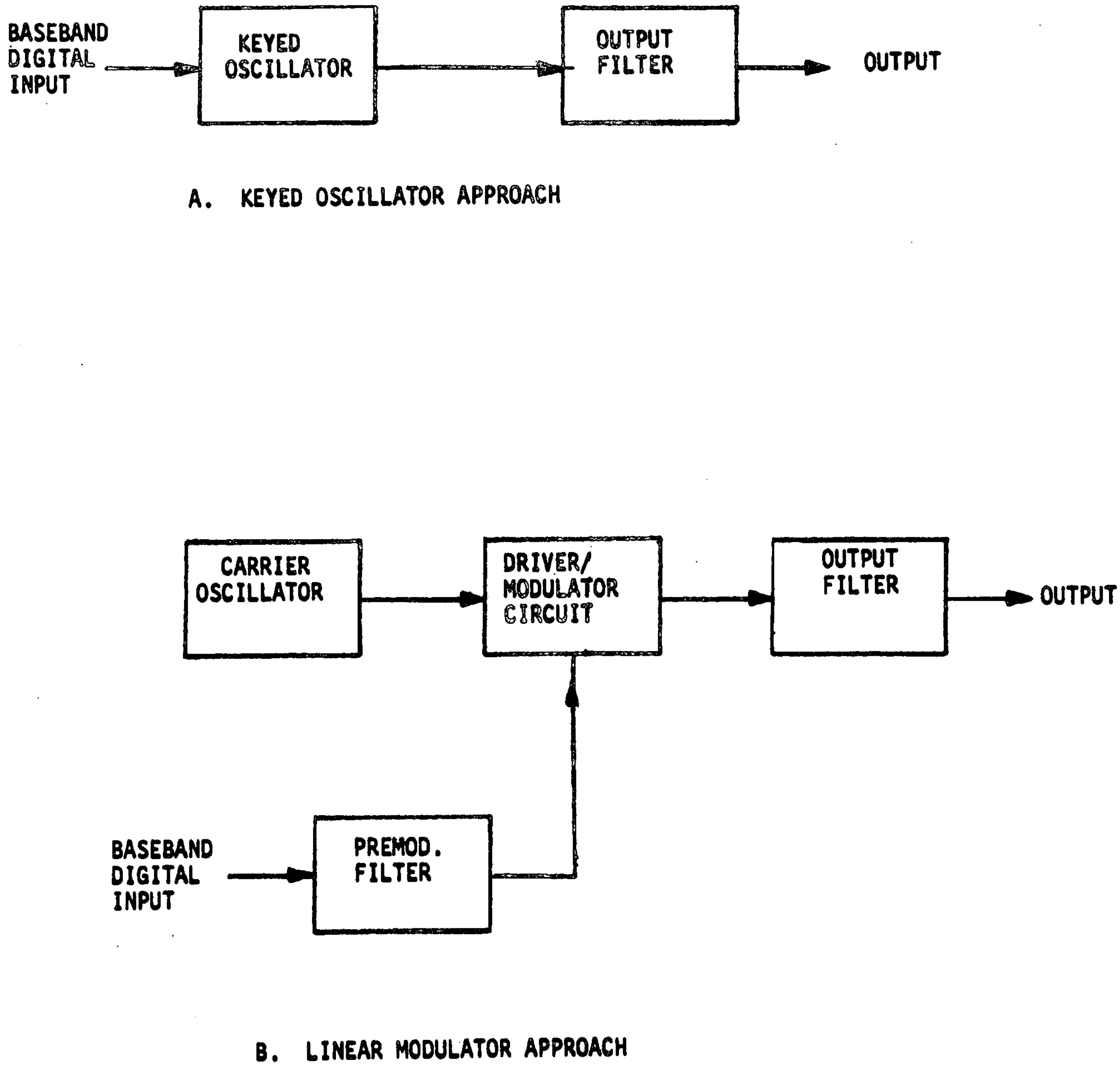


Figure 3-6. Modulation Technique Comparison



A. KEYED OSCILLATOR APPROACH

B. LINEAR MODULATOR APPROACH

Figure 3-7. ASK Double Sideband Modulators

minimizes harmonic distortion. Design is possible such that bandwidth requirements can be met even with single component failures in either the premodulation filter or the output filter. Two detection techniques are generally referred to in AM literature, simple envelope or noncoherent detection and product or coherent detection. Coherent detection can gain up to 3 db in signal-to-noise ratio but normally requires long acquisition times. Since the data bus system is not signal power limited but operates at a very fast asynchronous rate, noncoherent detection will be assumed for all AM techniques.

Vestigial sideband AM* is the technique employed in commercial TV and elsewhere to conserve bandwidth. It consists of suppressing or partially suppressing one of the sidebands so that the bandwidth approaches half that of standard AM. The common method of producing vestigial sideband is to uniquely design the output filter to pass the desired spectrum. This technique has the disadvantage of relying upon the output filter to prevent interference.

Receiver detector design is similar to that of standard AM except that only half the signal energy is available since one sideband is suppressed. Receiver filtering and equalization is related to the transmission spectrum and must be matched for an optimum signal output waveform. Thus the combined transmitter and receiver design is somewhat more sophisticated than simple AM and a fail-safe design more challenging.

Frequency Shift Keying (FSK)

FSK has the advantage of a constant power output independent of signal content. This is an advantage over ASK which contains a carrier that provides no signalling information. Hence for a given signal power, an FM output will be at least 50 percent lower than that of ASK. Other advantages of FSK are that performance under signal fading is good and signal-to-noise performance improvement is possible at the expense of additional bandwidth.

*Vestigial sideband is defined to be the technique of suppressing one sideband and part of the carrier. This general definition is sometimes called asymmetrical sideband with vestigial sideband being a special case.

Bandwidth of an FM system can be approximated by:

$$B = 2(\Delta F + fm) = 2\Delta F\ (1 + \frac{1}{D})$$

where

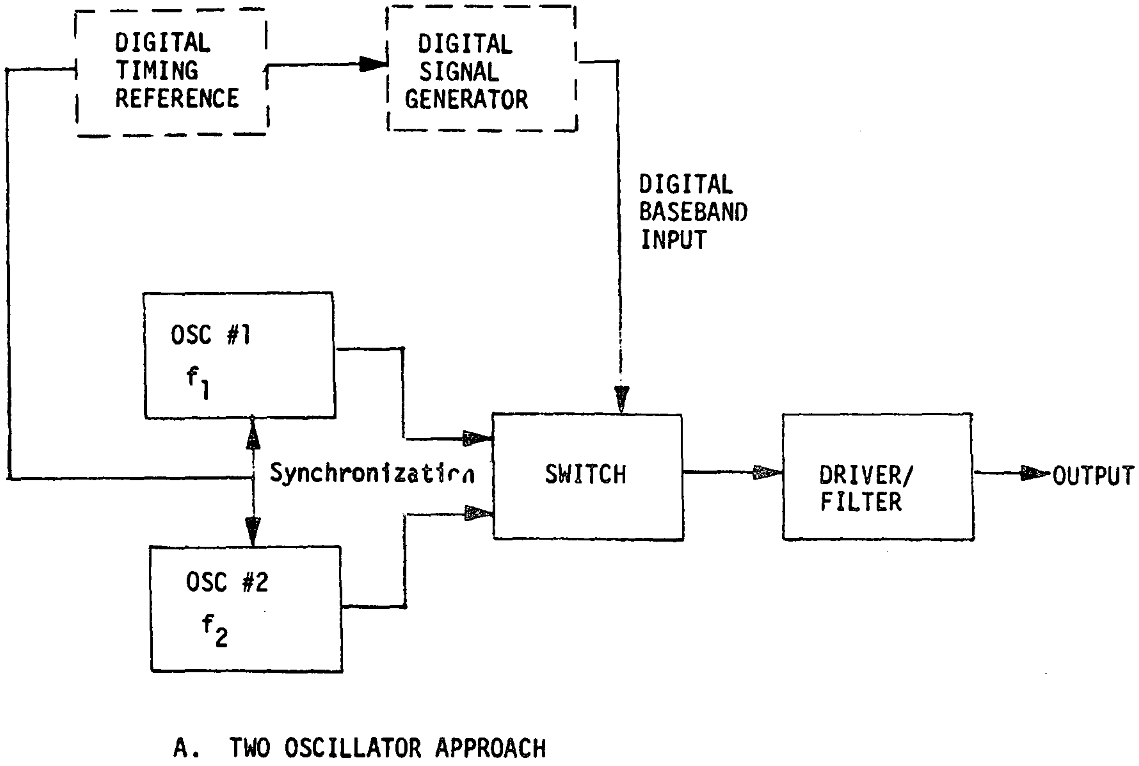
ΔF = frequency deviation

f_m = message bandwidth

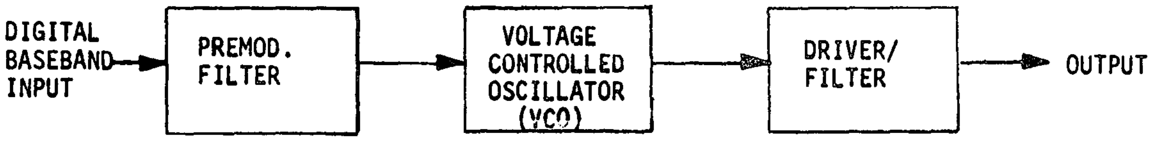
D = modulation index

For the case of narrow band FM, ΔF is made very small so that the net spectral bandwidth of the channel approaches $2f_m$ or equal to double sideband AM. A second method exists of reducing bandwidth in FSK without resorting to low modulation indexes. This approach involves permitting some amount of amplitude variation in the output waveform and consequently is not true FM. A limiter at the receiver can clip the signal and restore the waveform to the ideal FSK prior to detection.

Implementation of narrow band FSK would involve a highly stable frequency source and a carefully matched receiver design. The general problem in any FM design involving limiting bandwidths to that comparable to AM is minimizing discontinuities in the output waveform. This is illustrated in Figure 3-8. Separate oscillators are used for upper frequency, lower frequency and the digital data source. To avoid instantaneous phase changes in the output waveform at the switching points a synchronous relationship must exist between f_1 anf f_2 and digital switching to the output occurs when both sources have identical π or 2π phase relationships. This also implies synchronous digital signaling. Such an approach appears practical only at relatively low frequencies. Figure 3-8 shows the more conventional FSK approach. Problems in this design are centered about obtaining the required stability, fidelity, and linearity of the VCO. To maintain a restricted bandwidth, some degree of dependency on output filtering is mandatory. Hence a fail-safe design seems difficult to achieve.



A. TWO OSCILLATOR APPROACH



B. VCO APPROACH

Figure 3-8. FSK Modulator Implementation Techniques

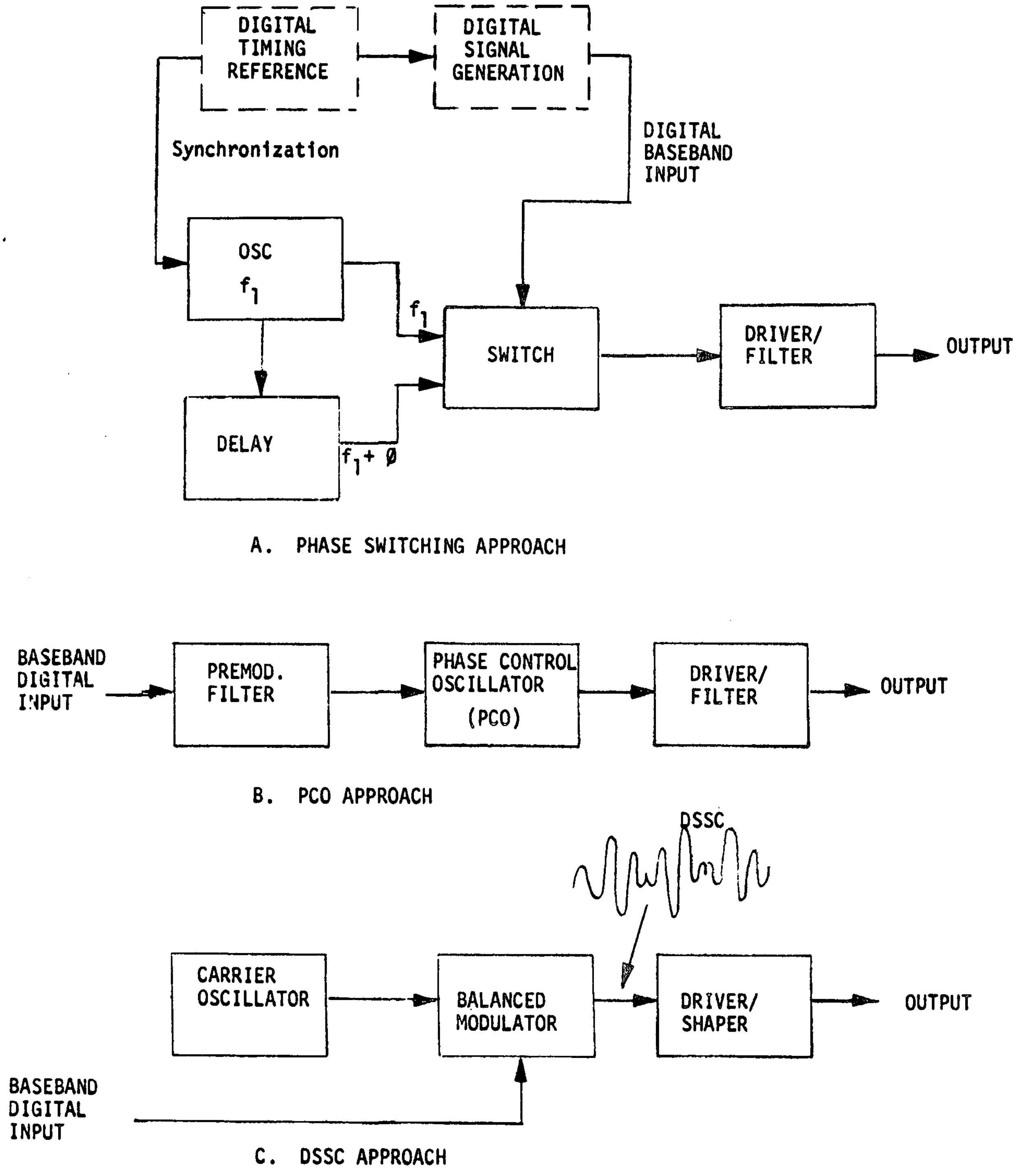
It would appear that conventional noncoherent FM receiver design techniques could be employed with the anticipated high signal-to-noise ratios. These include the discriminator, ratio detector or dual filter techniques (one for each of the binary signalling tones). In general the FM receiver would appear only slightly more complex than the AM envelope detector. Coherent FSK receivers requires reconstructing two ideal references (one for each signalling frequency), would be very sophisticated, and are not recommended for the data bus application.

Phase Shift Keying (PSK)

PSK, as noted earlier, is a nonlinear modulation with band spreading tendencies similar to FM. It has the advantage of optimum signal-to-noise and the constant output power property of FSK. However, because phase modulation is limited to maximum phase variations of $\pi/2$ without ambiguity, it does not offer the signal-to-noise improvement feature of FM in trading signal-to-noise for bandwidth.

Three modulator approaches, two of which are analogous to FSK, are shown in Figure 3-9. At higher frequencies the technique shown in Figure 3-9 would tend to be less practical for reasons similar to the analogous FSK case. Implementation of the approach of Figure 3-9 would be more applicable. Design complexity of this approach appears to be comparable with its FM analog. To minimize bandwidth to a spectrum comparable to AM, a minimum phase change could be used (analogous to narrow band FM) or as an alternate approach amplitude variation can be permitted. The later technique would place dependency on the output filter and again make a fail-safe design difficult. The third modulation scheme, Figure 3-9 permits amplitude variation by using a balanced modulator to develop a DSSC waveform. This technique takes advantage of the fact that DSSC and PSK with 180 degrees modulation rate have similar waveforms.

Coherent receiver approaches as for FSK are not practical because of the necessity of deriving a nearly perfect reference. However, the technique of detecting phase changes, called Differentially Coherent Phase Shift Keying (DPSK), is frequently employed and is not difficult to implement.



A. PHASE SWITCHING APPROACH

B. PCO APPROACH

C. DSSC APPROACH

Figure 3-9. PSK Modulator Implementation Techniques

Performance has been analytically shown to approach that of PSK at high signal-to-noise ratios. Figure 3-10 illustrates the detection technique.

PSK implementation appears to be comparable to FSK and somewhat more complex than ASK. Because of the band spreading tendancy, fail-safe designs are more difficult to achieve.

Table 3-3 summarizes the features of the principal candidate modulation techniques most important to data bus applications. The numbers in the table indicate the relative ratings of the techniques (no weighting factors applied) with best performance indicated by a one. A previously discussed, hardware complexity and fail-safe designs are of major importance.

Column 4 ratings are determined from Figure 3-6 performance curves for 10^{-6} error rates and operation in a Gaussian noise environment. Column 5 is the reciprocal of the bits per cycle of nominal bandwidth assuming that bandlimiting approaches to FSK and PSK are employed. Column 6 has not been elaborated on in the writeup but is included as being of noteworthy interest. Column 6 is derived from Bennett and Davey and is based on the average of tolerance to linear and parabolic delay distortions as defined in that reference.

The desire for simple designs, fail-safe features, reliable circuits, and the lack of signal-to-noise problems weigh heavily in favor of ASK. In addition, the wide usable bandwidth of coaxial cable should prevent bandwidth limiting. Consequently, conventional double sideband amplitude modulation will be used for the Space Station data bus digital system.

3.2.2.2 Timing and Synchronization

Timing and synchronization requirements are paritally established by the method of controlling bus access. The Space Station data bus employs polling or command/response techniques. Timing with this approach can be defined as the generation and/or distribution of the basic digital clocks required for

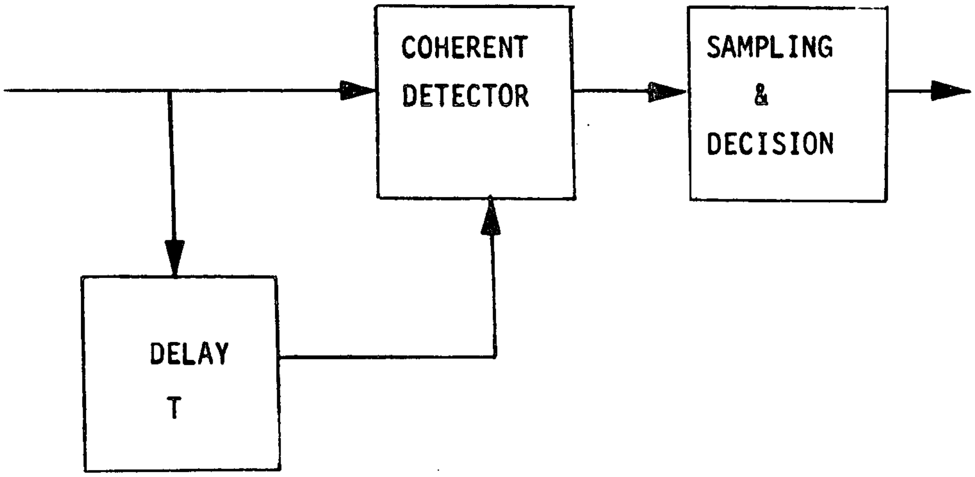


Figure 3-10. Differentially Coherent PSK Detection

Table 3-3

COMPARISON SUMMARY OF MODULATION TECHNIQUES FOR DIGITAL CHANNELS

	1	2	3	4	5	6
	Hardware Complexity	Ease of Fail Safe Design	Signal Power in Transmitter	S/N Performance	Reciprocal of Bits/ cycle of Nominal Bandwidth	Tolerance to Distortion
ASK (double side-band with carrier, 100% modulation)	1	1	2	3**	2	1
ASK (vestigial sideband, 100% modulation)	2	2*	3*	3**	1	3
PSK (differentially coherent)	2	2	1	1	2	2
FSK (non-coherent)	2	2	1	2	2	1

Notes

*Assumes bandwidth is controlled by output filtering
**Assumes less than optimum threshold detection
***Low number indicates best relative rating

logic operation. The system requirement to permit unit-to-unit communication also has an important influence on timing and synchronization. In digital communication, each receiver must resolve the optimum decision time for each data bit. If each unit on the bus communicates with only one other unit, simplifications using a central timing source are possible. However, when bus lengths are reasonably long and unit-to-unit communication is employed, alternate means must be employed to establish the decision time. This process of deriving a data clock is defined as part of the synchronization problem and discussed below, under the heading, "synchronization."

Timing

Two approaches to implementation are possible:

A. Centralized

B. Distributed

In the centralized timing scheme, a single oscillator is located in one of the units interfacing with the bus, and timing is distributed from it to all subsystems. This approach results in less hardware when the interface circuitry is less complex than multiple timing sources. It can also provide advantages in synchronous systems and as a vehicle for additional control (e.g., a change in clock amplitude can be used for synchronization or command). The functional disadvantage of the centralized approach is that subsystem operation ceases with the loss of the central clock channel. The recommended approach for the Space Station bus operation is distributed timing.

Hardware for transmitting clock from a central unit and receiving it in the DBTs involves somewhat greater complexity than simple crystal oscillators which meet stability requirements. In addition, a clock channel on the bus would require additional bandwidth (although not great), and is an additional source of interference due to potential harmonic distortion and cross modulation. Thus, unless overriding simplifications are possible due to synchronization or control, a distributed clock approach with separate oscillators in each DBT should provide the simpler and more failure tolerant approach.

The DBT to RDAU interface involves a physically short (about 10 to 20 feet) one Megaherz, serial digital communication link. In this case synchronous bit operation is possible, simple drivers and receivers are feasible, and a centralized clock timing distribution should yield overall hardware simplicity. Unless unique RDAU requirements indicate otherwise, a centralized timing system for the DBT to RDAU and vice versa transmissions are recommended.

Synchronization

Two types of synchronization are required for data bus operation:

A.. Message synchronization

B. Bit synchronization

Message synchronization involves techniques for identifying the start, end, and internal format check points within a message. Methods of implementation are numerous with the more positive techniques generally being the most difficult to implement. Candidates are as follows:

A. Amplitude synchronization—provide a signal of a unique level other than that used for data bits.

B. Pulse code synchronization—provide one or more data bits with known value or pattern.

C. Pulse width or frequency synchronization—provide one or more wider, or narrower pulses than normal.

The choice of waveform format influences message synchronization technique selection. That is, some methods of message synchronization are more compatible with one waveform than another. In general, amplitude synchronization will require an additional level detector, will tend to be bandwidth spreading, and consequently is not recommended.

Pulse code synchronization is independent of the digital format and involves a priori information. An example is the Barker code used in PCM telemetry in which a unique binary sequence is detected. Another simple special case of this technique is to use one bit (a predetermined one or zero). The obvious disadvantage of pulse code synchronization is that it is less "sure" than other techniques unless the code sequence is long. The advantages include independence of waveform format, no bandwidth constraint, and simplicity of implementation. Pulse width synchronization is the third technique. If narrower than normal pulses are used, greater channel bandwidth is necessary to pass synchronization. Wider than normal pulses do not require additional bandwidth but are not always possible depending on the format selection. The recommended technique will be discussed below, under the heading, "Message Synchronization."

Bit synchronization in an FDM system employing device-to-device transfer can be implemented in one of three ways as follows:

A. Transmit bit timing on a separate channel

B. Transmit bit timing within the same channel

C. Detect bit timing directly from data

The first approach of using a separate clock channel required additional bandwidth.transmitters and receivers and appears to offer no significant advantage.

The second technique suggests submultiplexing within a channel with frequency division submultiplexing the logical prime candidate. The minimum bandwidth for binary transmission is well known to be fs/2 where f_s is the signalling rate. Thus, since the clock is f_s, it is possible to transmit the clock and data on the same channel within a total message bandwidth of f_s. However, this approach requires careful filtering at both transmitter and receiver ends. Adjusting for filter delays and delay tolerances makes the approach unattractive. Thus, this approach is not recommended.

The third method, detecting bit timing directly from the data, is typically used in PCM telemetry and involves a synchronizer function to generate clock timing. As discussed in the next paragraph, the synchronizing function is simplified when numerous transitions occur in the data stream. Detecting bit timing from the data is the recommended approach.

Waveform Format Selection

There are numerous waveform format candidates for the Space Station data bus. Many of the more popular, simpler waveforms have been tabulated and defined in many references. Eleven of these are reviewed for the present application:

A. Polar NRZ level

B. Polar RZ

C. Bipolar NRZ

D. Dicode NRZ

E. Dicode RZ

F. Harvard code

G. Biphase L or Manchester II

H. Duobinary

I. Delay Modulation (Miller code)

J. Differential Ternary RZ (DTRZ)

K. 2790 Waveform

Trade parameters are obviated by noting the characteristics of an ideal waveform format. An ideal waveform for binary transmission will have the following properties:

A. Two signal levels

B. A bandwidth of fs/2 where fs is the signaling rate

C. No dc component (null in frequency spectrum at 0 frequency)

D. Regular predictable transitions

E. Process error detection properties

F. Will not tend to propogate errors

G. Be easy to implement

The first of these characteristics provides optimum signal-to-noise, requires only one threshold decision level and results in the simplest of receivers. The second represents the minimum possible bandwidth for two-level transmission. The third characteristic, no dc component, is a characteristic which simplifies and in some cases optimizes processing of the detected baseband signal. AC coupling simplifies receiver circuit designs. The fourth characteristic, regular predictable transitions, simplifies the generation of clock timing from the data. The fifth characteristic, processing error detection properties, involves a uniqueness of the waveform that is readily corrupted by additive noise. The sixth property, will not propogate errors, suggests that each data bit decision is independent of previous bits and/or that if bit synchronization is lost, multiple bit errors will not readily result. The last characteristic is a repeat of a previously stated objective.

It is easy to understand that no single waveform format can possess all of the above qualities and hence trades are required. For example, a two-level waveform with regular predictable transitions, if it is to convey information, must have more than the minimum fs/2 bandwidth. It is well known in communication theory that signal-to-noise can be traded for bandwidth. Thus, one method of obtaining many of the advantages above is to sacrifice the desirable two-level restriction and employ three levels. Many of the candidates under consideration are accordingly three-level waveforms that convey binary information.

Table 3-4 illustrates the candidate waveforms and identifies the desirable properties. The desired characteristics associated with each scheme are shown by checked blocks.

Of the waveform characteristics, the following criteria are considered most important.

A. Bandwidth should be no greater than fs. Since 10 megahertz bit rates are anticipated, waveforms involving greater circuit speeds would utilize excessive bandwidth and be proportionally more difficult to implement.

B. No dc component should be present. This property is especially important with ASK where the detected output is normally biased at a level that is a function of signal strength and percent modulation. It also permits pulse width message synchronization without increase in bandwidth.

C. The waveform should have error detection properties. Error detection properties can be a powerful tool in detecting burst error noise.

D. Implementation should be relatively simple (rating of 2 or lower in table).

With the above selection criteria, all waveforms are eliminated except bipolar NRZ, dicode NRZ, dicode RZ and biphase L. Dicode RZ offers only a slight advantage* over dicode NRZ and requires twice the bandwidth. Therefore it is also eliminated. The differences between bipolar NRZ and dicode NRZ are subtle with each requiring a bandwidth of fs/2. With this observation, recommendations for the Space Station application are:

A. Biphase L if channel bandwidths of fs (10 MHz) are practical

B. Bipolar NRZ or dicode NRZ if bandwidths must be limited to fs/2.

Biphase L is the preferred approach because it provides numerous transitions for ease of clock generation, and improved signal-to-noise performance by virtue of being a two-level system. If bandwidth restrictions necessitate the

*Less intersystem interference

Table 3-4

SIGNAL WAVEFORM STUDY

	SIMPLICITY	NUMBER OF LEVELS	NO DC COMPONENT	BAND-WIDTH	CLOCK SYNC	ERROR DETECTION	PROPOGATES ERRORS
POLAR NRZ-Level	1	✓ 2	-	✓ $f_{s/2}$	-	-	✓
POLAR RZ	1	- 3	-	- f_s	✓	✓	✓
BIPOLAR NRZ	2	- 3	✓	$f_{s/2}$	-	✓	✓
DICODE NRZ	2	- 3	✓	$f_{s/2}$	-	✓	-
DICODE RZ	2	- 3	✓	- f_s	-	✓	-
HARVARD CODE	1	- 3	✓	-	✓	✓	✓
BIPHASE LEVEL (Manchester II)	2	✓ 2	✓	- f_s	✓	✓	-
DUOBINARY	(3)	- 3	-	✓ $f_{s/4}$	-	✓	
DELAY MODULATION (Miller Code)	3	✓ 2	-	✓ $f_{s/2}$	✓	-	-
DIFFERENTIAL TERNARY RZ	1	- 3	✓	- $2f_s$	✓	✓	-
2790 WAVEFORM	1	- 3	✓	2f or $^{3f}/_2$	✓	✓	✓

1 = MOST SIMPLE

use of bipolar NRZ or dicode NRZ, it is recommended that parity bits be liberally used in the format to insure frequent transitions regardless of data content. It should be noted that bipolar NRZ or dicode NRZ will require a more sophisticated synchronizer bit generator such as a phase lock loop circuit.

Message Synchronization

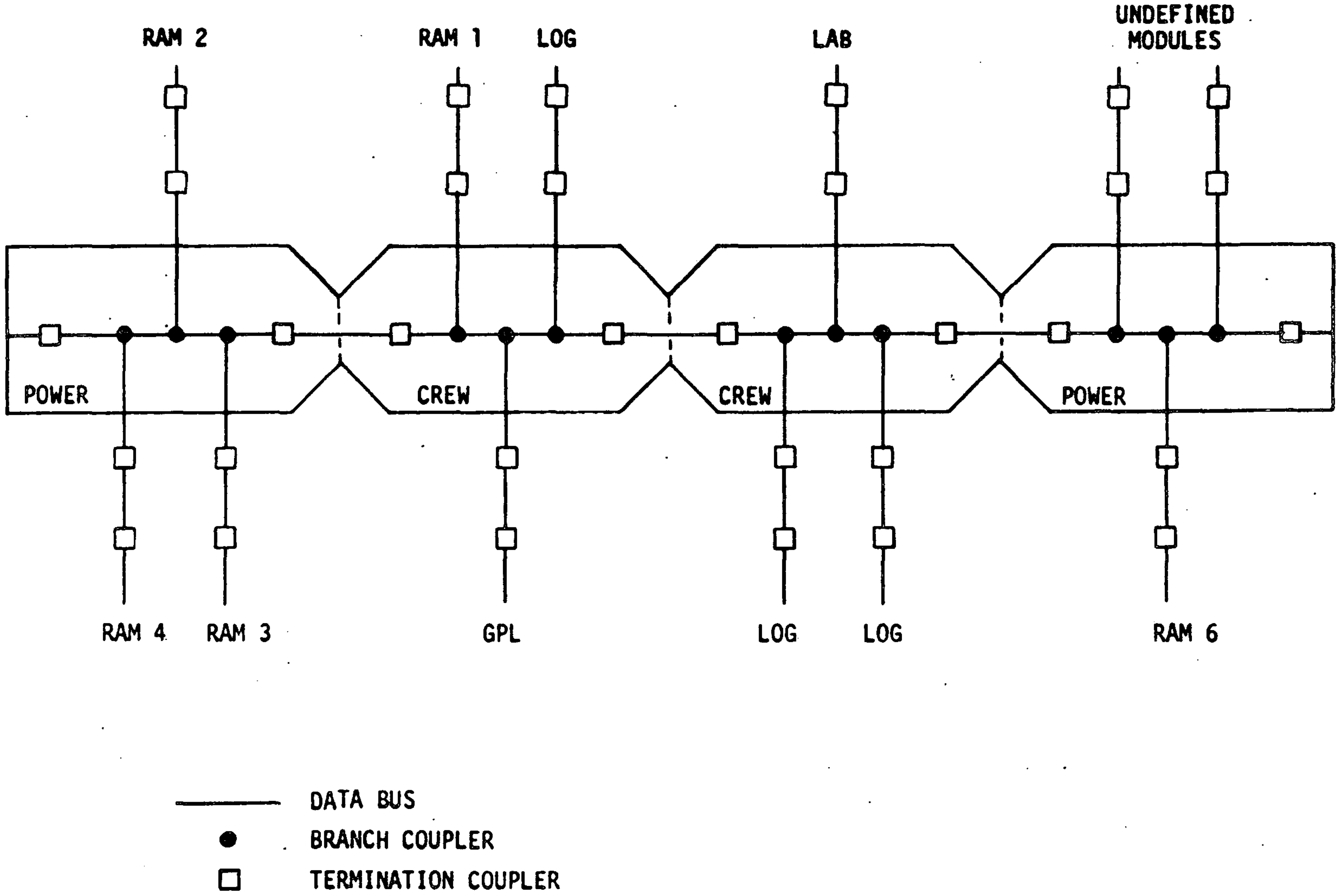
Since a biphase L code will be used in all message transactions, a string of all one's or all zeros will result in the maximum number of transitions. Consequently, it is recommended that a string of at least 15 zeros be sent as a synchronization burst. This will allow 1.5 microseconds for the clock circuits to acquire synchronization and be properly phased. The phase can be forced since the sequence is known to be all zeros. To insure against false starts, the final three bits of the synchronization burst (to fit the 18 bit word format and ease timing problems) can be coded such as one, zero, one combination. Since all synchronization bursts are followed by an "A" word, these characteristics can be checked by logic to prevent false starts, erroneous decoding, and noise actuated message responses.

3.2.3 Signal Distribution

A block diagram of the proposed Space Station digital data distribution system is shown in Figure 3-11. The main bus length is approximately 200 feet with each branch line not more than 50 feet in length. The total number of digital terminals tied to the bus is a maximum of 128 with 12 branches. These terminations must be made through the use of couplers.

3.2.3.1 Coupling Analysis

Any mismatch occurring at any point along the data bus will cause energy to be reflected back along the line with a resulting loss of energy in the incident signal. The reflections behave in exactly the same manner as the incident signal and cause echoes or ghosts on TV images and false triggering of digital devices. The cable used in the data bus must therefore be matched; i.e., all devices connected to the line must possess an impedance equal to the characteristic impedance of the cable.



NOTE: Same connection concept for both Analog and Digital Buses.

Figure 3-11. Data Buses Connection Diagram

The devices currently used to couple energy to and from a coaxial transmission line include:

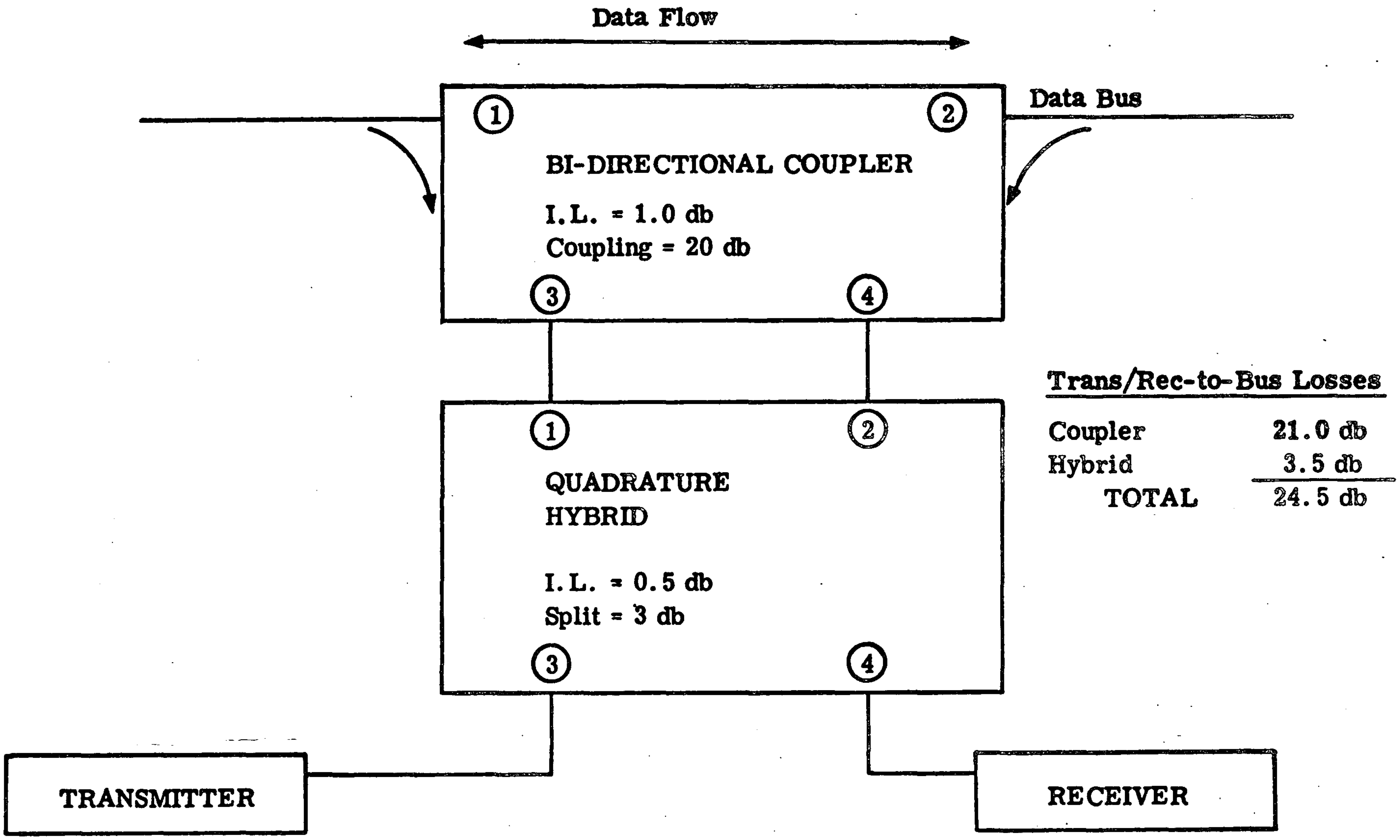
A. Directional couplers
B. Quadrature hybrids
C. Hybrid junctions
D. Power splitters
E. Resistive and capacitive networks
G. Transformers
G. Pressure taps
H. Tees

In order to properly design a balanced, i. e., matched, distribution system, the many types of couplers and their characteristics have to be investigated before the data transmission line can be completely designed.

Directional Couplers

Directional couplers are generally not suitable for data bus applications since they will couple energy from a transmission line in only one direction; i. e., they are not bi-directional. However, a directional coupler can be made bi-directional by modifying the coupler to bring the fourth port (normally internally terminated) to the coupler exterior as shown in Figure 3-12. Coupled port 3 receives energy from port 1 while port 4 receives energy from port 2. The amount of coupling can be any desired value and generally is in 10 lb increments from 10 to 40 db. A quadrature hybrid is used to split energy at any port equally between the opposite pair of ports and isolate the input/output port from the adjacent input/output port. Thus, when the transmitter is operable, the adjacent receiver has a high degree of isolation.

The primary disadvantage of the scheme of Figure 3-12 is that in the vicinity of 300 MHz, the insertion loss approaches 1 db. Assuming an end-to-end data transmission (i. e., a transmitter at one end of the bus communicating with a receiver at the far distant end), the total signal loss would be approximately 182 db.



Trans/Rec-to-Bus Losses

Coupler	21.0 db
Hybrid	3.5 db
TOTAL	24.5 db

Figure 3-12. A Bi-Directional Coupler Configuration Using Directional Coupling and A Quadrature Hybrid

Assuming a practical upper limit of transmitter power of one watt (30 dbm), the required receiver sensitivity is

$$\begin{aligned} \text{Receiver sensitivity} &= P_T - \text{losses} \\ &= +30 - 181.8 \\ &= -151.8 \text{ dbm} \end{aligned}$$

This value is beyond the present capabilities of receivers.

Quadrature Hybrids

Quadrature hybrids are similar to the couplers previously described in terms of loss characteristics, bandwidths and impedance levels. Quadrature hybrids, Figure 3-13, operate as follows:

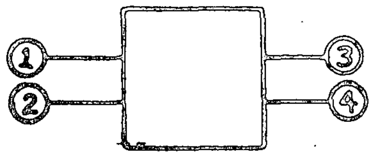


Figure 3-13. Quadrature Hybrid Representation

Energy at port 1 will divide equally between ports 3 and 4 and there will be practically no energy present at port 2. The phase of the signals at the ports 3 and 4 will be 90 degrees apart. The quadrature hybrid is a symmetrical device; i. e., a signal injected at any port will split between the opposite two ports and will be isolated from the adjacent port when all ports are terminated in the characteristic impedance of the hybrid.

Hybrid Junctions

Hybrid junctions cover a variety of devices and have been given various names such as magic tees, rat races, ring hybrid, and hybrid tees. The hybrid junction operates as follows.

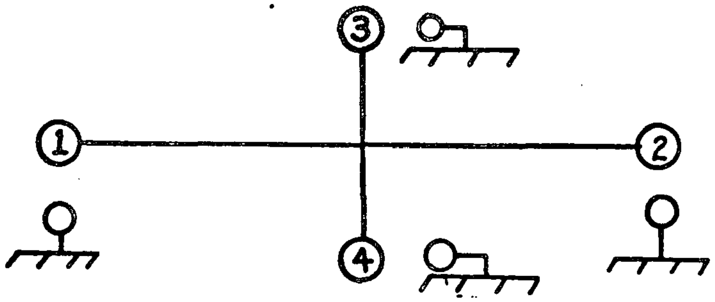


Figure 3-14. Hybrid Junction Representation

In Figure 3-14 a signal applied to port 1 splits equally between ports 3 and 4 with an 0 degree shift between them. Signals applied to port 2 will split equally between ports 3 and 4 with 180 degrees phase shift between them. Ports 1 and 2 are isolated from each other. Simultaneous signals applied to ports 1 and 2 will combine such that their vector sum will appear at port 3 and their vector difference will appear at port 4.

The hybrid junction losses, bandwidths, and costs are comparable to the directional coupler.

Pressure Tap

The pressure tap was developed for use primarily in the community area television (CATV) industry where connections are made to a coaxial cable by inserting connectors. In addition to the ease of a coax cable connection, the pressure tap connection is bi-directional and appears to have the lowest amount of loss of any coupling device with the possible exception of resistive and/or capacitive networks. A typical pressure tap used largely in the CATV industry is the Jerrold Electronics Model CMT. The specification of this device are:

Impedance	75 ohms
VSW	1. 3:1 Maximum at 216 MHz
Frequency	to 800 MHz

Coupling Loss (db)	Insertion Loss (db)
36	0. 07
30	0. 07
24	0. 15
20	0. 25
16	0. 4
12	0. 7

Power Splitters

Power splitters provide a convenient method of dividing a signal from a signal source to two separate loads while maintaining a matched impedance condition and providing isolation between the loads. As shown in Figure 3-15, the power splitter is not applicable to insertion directly onto the data bus line since this device is undirectional. Energy appears at port 3 only when signals enter port 1. A signal entering port 2 is not coupled to port 3. In addition, energy cannot be coupled from port 3 to port 2. The power splitter finds application "off" the main data bus line as shown in Figure 3-16.

Resistive and Capacitive Networks

Resistive and capacitive networks may be used as signal samplers and couplers to coaxial lines and design of these devices follows general network circuitry theory. One such device used in television distribution systems is a capacitive network in a printed wiring board configuration. This unit is rather large and heavy and is presently only in limited use. Typical insertion losses are 0.4 db for 15 db coupling loss and 1.5 db for 10 db coupling loss. Resistive signal samplers can be designed for wide bandwidth and low VSWR. However, coupling device data indicate that for a 20 db resistive coupler, an insertion loss of approximately 1 db can be expected over the frequency range of a few megahertz to 300 MHz.

As the coupling losses increase, the insertion loss decreases. Therefore, for a large number of terminations, a larger coupling loss can be offset by the savings in insertion loss.

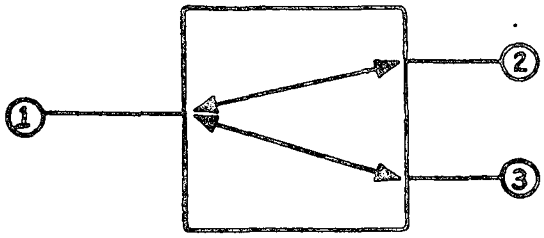


Figure 3-15. Power Splitter Representation

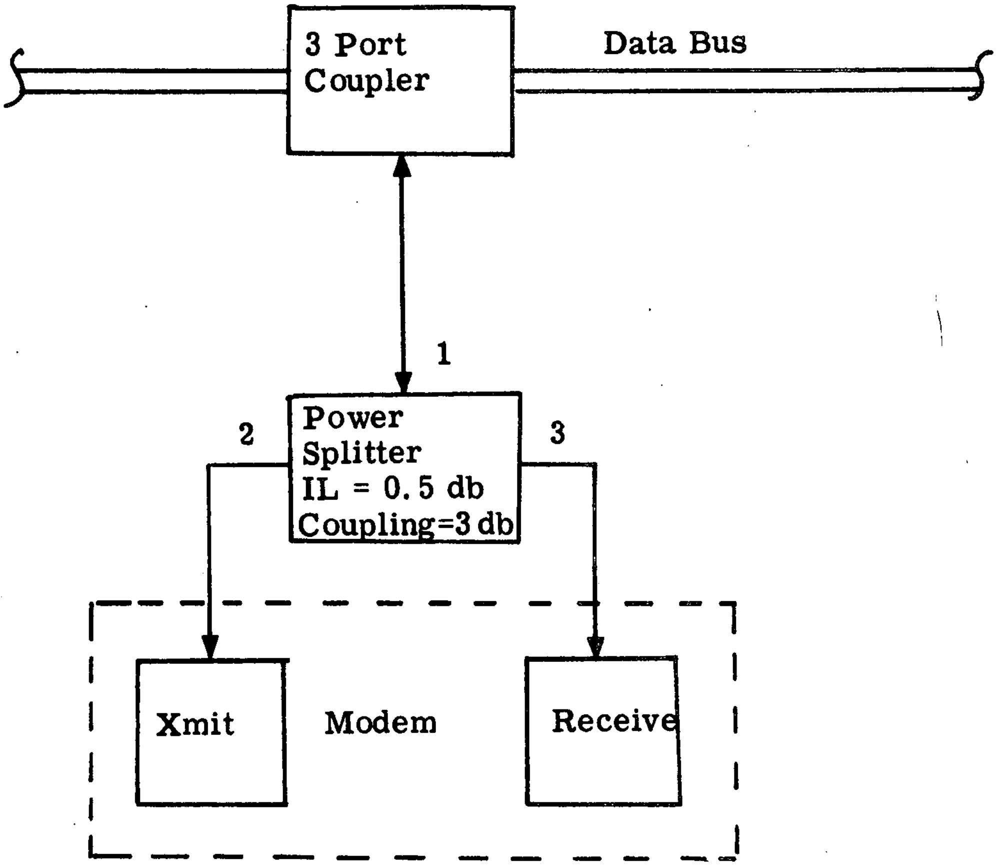


Figure 3-16. Power Splitter Application

The recommended couplers for Space Station terminations are discussed in the following paragraphs.

3.2.3.2 Data Bus System Signal Power Analysis

The design of the specific systems interfacing with the data bus depend on the assigned frequencies, system receiver capabilities, and number of terminations to be serviced. These factors, in turn, depend on the basic concept of either one data bus or the use of multiple data buses. For the purpose of analysis, a coaxial cable similar to 3/8 inch Alumiform, 75 ohm cable is assumed in either case. At 20°C ambient temperature, the loss at 300 MHz is 2.0 db per 100 ft of cable. The resultant power budget assumptions are shown in Table 3-5 for those factors common to both a one bus and a multiple bus system. Table 3-6 contains those characteristics which are different in the two concepts.

One Bus Concept Analysis

For the purpose of this analysis, the core modules consisting of two power modules and two crew modules are considered to be the main data bus line. All other modules, e.g., a RAMS or logistics module, are connected to the

Table 3-5
COMMON POWER BUDGET ASSUMPTIONS

Coaxial Cable Loss	2 db/100 ft at 150 MHz 1 db/100 ft at 70 MHz 0.3 db/100 ft at 1 MHz
Digital RF Switch Insertion Loss	1.5 db
Receiver Characteristics	
Digital N.F. = 8 db, B.W. = 25 MHz	
Video N.F. = 10 db, B.W. = 4.2 MHz (6 MHz vestigial)	
Audio N.F. = 25 db, B.W. = 3 KHz	
Required Detected Signal to Noise Ratios	
Digital	22 db
Video	39 db
Audio	15 db

Table 3-6

COUPLER CHARACTERISTICS

One Bus			
	Digital	at 150 MHz	17.0 db coupling
			0.11 db insertion loss
		70 MHz and below	30 db coupling
			Negligible insertion loss
	Video broadband		26 db coupling
			0.3 db insertion loss
	Audio broadband		40 db coupling
			0.5 db insertion loss
	Branch point broadband		16 db coupling
			0.5 db insertion loss
	GPL module branch point		7.0 db coupling
			2.0 db insertion loss
Multiple			
	Digital		17.0 db coupling
			0.11 db insertion loss
	Analog		
	Video		20 db coupling
			0.2 db insertion loss
	Audio		35 db coupling
			0.1 insertion loss
	Branch point broadband		12 db coupling
			1.0 db insertion loss

data bus with the broadband branch port connector. The GPL Module is connected to the bus with a special branch port connector. The assumed distribution of terminations is shown in Table 3-7.

The power budgets of each of the three systems are shown in Tables 3-8, -9, and -10. It can be seen for a one bus system with 96 main line terminations plus 12 branch points that the required audio transmitter power is 25 dbm, the required video transmitter power is 38 dbm, and the required digital system transmitter power is 30 dbm.

The intersystem interference power levels are shown in Table 3-11. It can be seen that the only system requiring attention from intersystem interference is the audio receive system. If the audio transmitter power is increased from 25 to 30 dbm with a corresponding decrease in receiver sensitivity to -105 dbm, the corresponding interference level from an adjacent digital transmitter will be decreased to 7. 5 db above threshold and 12. 5 db below worst case signal level. This corresponds to better than 90 percent intelligibility.

Table 3-7
TERMINATION DISTRIBUTION

One Bus	
Digital	64 Main line data bus 8 Worst case branch
Video	16 Main line data bus 3 Worst case branch
Audio	12 Main line data bus 3 Worst case branch
Multiple Bus	
Digital	64 Main line 15 Branch line
Video	12 Main line 4 GPL branch 2 Branch
Audio	12 Main line module 4 GPL 3 Branch

Table 3-8

POWER BUDGET

Digital System	
Power Loss	
Coaxial Cable	4.8 db
Main Line Termination	12.5
Branch line termination	3.4
Branch points	7.5
Transmit coupling	18.5
Receive coupling	18.5
2 branch point couplings	23.0
	88.2 db
Expected VSWR Loss	4.8
Total Loss	93.0 db
Receiver Sensitivity	-92 dbm
Required S/N	22 db
Required received power	-70 dbm
Required modem transmit signal power	+23 dbm
Required unmodulated carrier transmitter power	+30 dbm

However, interference problems become apparent when the digital system operation is analyzed in depth. The characteristics from Table 3-8 are utilized in conjunction with the modem characteristics, Table 3-12, to derive the figures shown in Table 3-13. When the two adjacent digital channels are placed at 140 MHz and 180 MHz, the third harmonic power from the adjacent channel falls within the pass band. Consequently, only transmitter filtering provides adjacent channel immunity to third harmonic interference. The results are intolerable since a signal to interference power ratio of more than 10 db is required to maintain the bit error rate system design goal of 10^{-7}.

Table 3-9
POWER BUDGET

Video System	
Power Loss	
Coaxial Cable	2.4 db
Main line terminations	5.6 db
Branch line terminations	1.7
Branch points	7.5
Transmit coupling	26.0
Receive coupling	26.0
Branch point coupling	23.0
	92.2
Expected VSWR Loss	< 4.5
Total Loss	96.7 db
Receiver Sensitivity	-98 dbm
Required S/N	39 db
Required Receiver Power	-59 dbm
Required Transmitter Power	38 dbm

Another major problem becomes apparent when the adjacent channel transmitter fundamental and carrier powers are considered to be co-located with a receiver. Table 3-14 illustrates the problem. When the "next door" transmitter is on, the receiver trying to obtain data from a distant transmitter will be completely swamped. Further, it becomes apparent from Table 3-14 that all adjacent channel systems co-located will cause distinct and insurmountable interference problems. In order to alleviate this problem, separate buses are required.

Table 3-10

POWER BUDGET

Audio System	
Power Loss	
Coaxial cable	1.0 db
Main line terminations	5.6
Branch line terminations	1.7
Branch points	7.5
Transmit coupling	40.0
Receive coupling	40.0
Branch point coupling	23.0
	118.8
Expected VSWR Loss	< 1.3
Total Loss	$\boxed{120.1 \text{ db}}$
Receiver Sensitivity	-110 dbm
Required S/N	15 db
Required Received Power	$\boxed{-95 \text{ dbm}}$
Required Transmitter Power	$\boxed{+25 \text{ dbm}}$

Multiple Bus Concept

Because of interference problems consider two buses, an analog bus connecting all audio and video stations and one connecting all digital terminations. Since the digital system design was most critical for a one-bus data distribution system, that particular design will be analyzed prior to the analog bus. Since only digital transmission of 140 MHz or higher are on this bus, the coupler used in the design of the one-bus digital system can be utilized for branch port connections, also. The digital terminations remain the same, 64 main line and a worst case of 8 per branch. The power budget of the system is shown in Table 3-15. It can be seen that the required transmitter power has dropped 10 db from that of the one-bus digital transmitter.

Table 3-11

INTERSYSTEM INTERFERENCE LEVELS

I. Video Transmit - Digital Receive	
Transmit power	38 dbm
Video coupler	-26 db
Digital coupler	-31.5
Digital receiver filter	-75.0
Received Interference Power	-95.5 dbm
Summary	
Below threshold	2.5 db
Below signal level	24.5 db
II. Digital Transmit - Video Receive	
Transmit power	+30 dbm
Digital coupler	-18.5 db
Video coupler	-26.0
Video receiver filter	-75.0
Received Interference Power	-88.5 dbm
Summary	
Above threshold	8.5 db
Below signal level	30.5 db
III. Video Transmit - Audio Receive	
Transmit power	38 dbm
Video coupler	-26 db
Audio coupler	-40 db
Audio receiver filter	-75
Received Interference Power	-104 dbm
Summary	
Above threshold	7 db
Below signal level	13 db
IV. Digital Transmit - Audio Receive	
Transmit power	+30 dbm
Digital coupler	-18.5 db
Audio coupler	-40.0 db
Audio receiver filter	-75.0
Received Interference Power	-102.5 dbm
Summary	
Above trheshold	12.5 db
Below signal level	7.5 db

Table 3-12
MODEM CHARACTERISTICS

Receiver	
Sensitivity	MINUS 92 dbm
Bandwidth	25 MHz
Out-of-band rejection	$f_c \pm 30 \geq 15$ db
	$f_c \pm 50 \geq 45$ db
	$f_c \pm 60 \geq 70$ db
	$f_c \pm 70 \geq 75$ db
Transmitter	
Unmodulated carrier power	31 dbm
Modulation wave third harmonic power output	≤ MINUS 23 dbm

Table 3-13
ADJACENT CHANNEL INTERFERENCE

Critical Channels	
Digital 10 MBPS at 140 MHz and 180 MHz	
Third Harmonic Interference	
180 - 30 = 150 MHz	
Transmit Power	-23 dbm
Transmit coupler	-18. 5
Receive Coupler	-18. 5
Received Interference Power	$\boxed{-60 \text{ dbm}}$
Summary	
Above threshold	32 db
Above signal level	10 db

Table 3-14

ADJACENT CHANNEL INTERFERENCE

Critical Channels	
Digital 10 MBPS at 140 MHz and 180 MHz	
Fundamental and Carrier Interference	
Transmit power	30 dbm
Transmit coupler	-18.5 db
Receive coupler	-18.5
Receive filter attenuation	30.0
Received interference power	-37 dbm
Summary	
Above threshold	50 db
Above signal level	22 db

Table 3-15

SEPARATE DIGITAL BUS POWER BUDGET

Power Loss	
Coaxial cable	4.8 db
Main line termination	7.1
Branch line termination	1.7
Branch points	12.0
Transmit coupling	18.5
Receive coupling	18.5
2 branch point couplings	24.0
	79.5
Expected VSWR loss	3.4
Total Loss	82.9 db
Receiver Sensitivity	-92 dbm
Required S/N	22 db
Required Received Power	-70 dbm
Required Modem Transmit Signal Power	13 dbm
Required Unmodulated Carrier Transmitted Power	20 dbm

Since the buses are now separated, only interference from digital channels need be considered in the choice of channel frequencies. Three digital channels can be located at carrier frequencies less than or equal to 300 MHz by locating at 140 MHz, 210 MHz and 300 MHz. (A carrier frequency much less than 140 MHz for a 10 MBPS biphase signal represents too much percentage of modulation bandwidth to be practical.) Therefore, the adjacent digital channel interference levels become negligible as shown in Table 3-16 with a separate digital data bus. Other advantages to this system design also become apparent in the analog design.

The video system power budget is shown in Table 3-17 and the audio system power budget in Table 3-18. The required video transmitter power is 28 dbm compared with the 38 dbm required for one bus operation. The required audio transmitter power is 21 dbm. It can be seen that branch to branch communications is possible for both systems while maintaining the required signal to noise ratios.

The intersystem interference levels on the analog bus are shown in Table 3-19. These levels do not degrade either system operations in any manner.

Table 3-16

SEPARATE DIGITAL BUS ADJACENT CHANNEL INTERFERENCE

Transmit power	20 dbm
Transmit coupler	-18. 5 db
Receive coupler	-18. 5
Receiver filter attenuation 140 MHz f_c to 210 MHz f_c	-75
Received interference power	$\boxed{-92 \text{ dbm}}$
Summary	
Above threshold	0 db
Below signal level	22 db

Table 3-17

ANALOG BUS POWER BUDGET

Video System	
Power Loss	
Coaxial cable	2.4 db
Main line terminations	3.6 db
Branch line terminations	1.9
Branch points	12.0
Transmit coupling	20.0
Receive coupling	20.0
Branch point coupling	24.0
	83.9
Expected VSWR Loss	2.6
Total Loss	86.5 db
Receiver Sensitivity	-98 dbm
Required S/N	39 db
Required Receiver Power	-59 dbm
Required Transmitter Power	+28 dbm

3.2.4 Frequency Selection

Frequency selection was originally performed on the basis of an integrated analog and digital bus. The major concern at this time was to choose frequencies such that their intermodulation products would not appear in the selected frequency passbands. The necessity to revert to dedicated analog and digital buses resulted in the previous work being of academic interest only. However, it will be included for completeness. Since there are only three frequencies now appearing on the digital bus, the analysis to determine satisfactory spectrum locations becomes simple enough to be hand calculated.

3.2.4.1 Dedicated Digital Frequency Analysis

The result of passing frequencies through a non-linear device results in a response of the form

Table 3-18
ANALOG BUS POWER BUDGET

Audio System	
Power Loss	
Coaxial cable	1.0 db
Main line terminations	1.2
Branch line terminations	0.7
Branch points	12.0
Transmit coupling	35.0
Receive coupling	35.0
2 branch point couplings	24.0
	108.9
Expected VSWR loss	21.3
Total Loss	110.2 db
Receiver Sensitivity	-105 dbm
Required S/N	15 db
Required Received Power	-90 dbm
Required Transmitter Signal Power	+21 dbm

$$E_o = \sum_{j=1}^{m} \sin W_u t + \left[\sum_{j=1}^{m} \sin W_j t \right]^2 + \left[\sum_{j=1}^{m} \sin W_j t \right]^3$$

m = number of carriers neglecting higher order low amplitude responses.

For m equal to three, only three products are generated of the second order and only six of the third. Since the first carrier was established at 140 MHz as a reasonable lower bound and the second set at 210 MHz in order to insure sufficient attenuation of an adjacent frequency by the receiver filter, only the value of the third frequency is in question.

A trial frequency of 280 MHz resulted in an intermodulation product within a passband but a trial of 300 MHz was successful although products do appear

Table 3-19

ANALOG BUS INTERSYSTEM INTERFERENCE LEVELS

I. Video Transmit - Audio Receive	
Transmit power	28 dbm
Video coupler	-20 db
Audio coupler	-35 db
Audio receiver filter	-75 db
Receiver interference power	-102 dbm
Summary	
Above threshold	3 db
Below signal level	17 db
II. Audio Transmit - Video Receive	
Transmit power	21 dbm
Video coupler	-30 db
Audio coupler	-35 db
Video receiver filter	-75 db
Received interference power	-119 db
Summary	
Below threshold	21 db
Below signal level	60 db

at 300 ±20 MHz. However, they are third order products with amplitudes one-fourth of the original carriers and since they are multiplied by the magnitude of the non-linearity which should not exceed 10 percent, their effect on the system is negligible.

3.2.4.2 Integrated Frequency Analysis

Assumptions

Want 3 PCM's

Each PCM = 10 Megabits

Assume Biphase PCM symbols as follows (Figure 3-17)

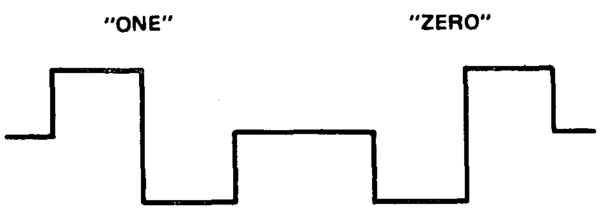


Figure 3-17. Biphase PCM Symbols

Determine spectrum of Biphase PCM. Obtain PCM bandwidth from Proceeding of IEEE, July 1969, page 1316. (Figure 3-18)

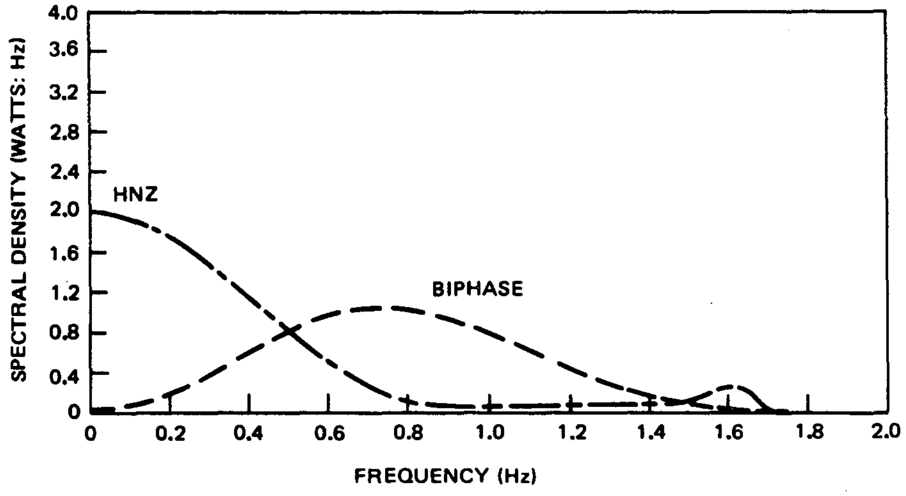


Figure 3-18. Spectrum of Biphase PCM

Read PCM bandwidth to be 12. 5 Megahertz.

Let PCM modulate a carrier, (either A. M. or narrow band F. M.). Then bandwidth of each PCM channel = 25 megahertz.

Bandwidth of a PCM channel = 25 MHz per channel
x3 Channels
75 MHz for 3 channels

Let Guard band = 20 MHz between channels
Total of guard bands = 2 x 20 = 40
Spectrum of 3 channels of PCM = 40 + 75 = 115 MHz

To eliminate 2nd order products assume: "Highest frequency subcarrier should be less than twice the lowest subcarrier frequency."

So let lowest = 115 MHz
and highest = 230 MHz.

Try:	PCM #1	140 MHz
	#2	180 MHz
	#3	260 MHz

Put PCM's on same line with analog channels, add non-linearity, and compute intermodulation noise in each channel.

Define non-linearity to be 2nd and 3rd order per Figure 3-19.

Calculate intermodulation noise per "D SMSPC" Computer program and tabulate results.

Referring to Table 3-20, the intermodulation noise present in a channel is proportional to percent 2nd and 3rd order non-linearity.

If the squared and cubed terms are held each to 1.0 percent, the signal to noise ratio will be better than 40 db. If non-linearity is 10 percent, S/N is about 25 db. If non-linearity of 0.1 percent would be about 60 db S/N. PCM sould work well with 40 db S/N. Analog functions, such as TV, may require 60 db.

If 0.1 percent non-linearity is required, referring to Figure 3-19, summarize a few of the voltage values in Table 3-21.

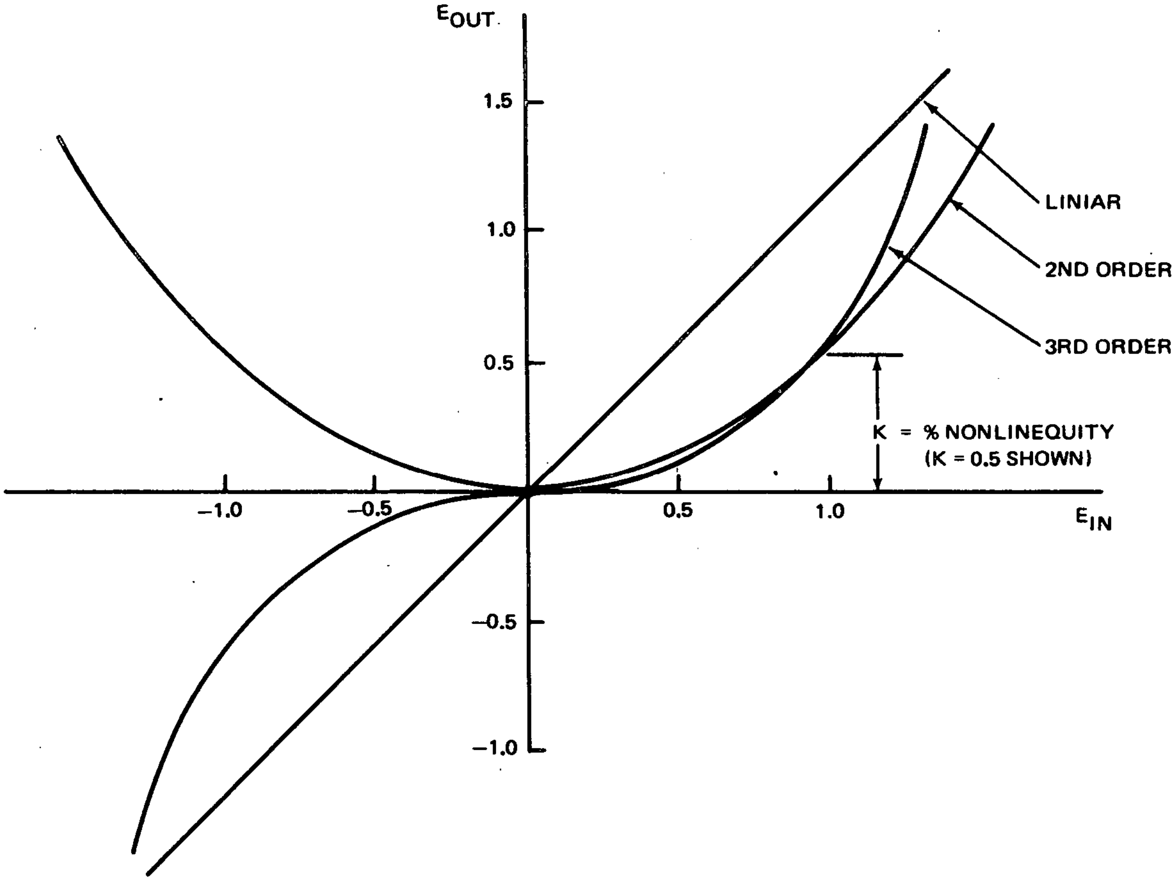


Figure 3-19. Nonlinearity Description

Table 3-20

NOISE DUE TO NONLINEARITY

Function	Frequency Megahertz	Signal Milli-Volts	Intermodulation Noise in Band			
			k = 10%		k = 1%	
			MV	S/N	MV	S/N
PCM 1	247.5 to 272.5	58.9				
PCM 2	167.5 to 192.5	58.9				
PCM 3	127.5 to 152.5	58.9	2.4	28 db	0.24	48 db
0.8 Test	70.75 to 76.75	58.9	2.6	26 db	0.26	46 db
TV 1	62.75 to 68.75	58.9				
TV 2	54.75 to 60.75	58.9				
TV 3	46.75 to 52.75	58.9				
TV 4	38.75 to 44.75	58.9				
TV 5	30.75 to 36.75	58.9	3.3	25 db	0.33	45 db
TV 6	22.75 to 28.75	58.9	3.6	24 db	0.36	44 db
TV 7	14.75 to 20.75	58.9	3.9	24 db	0.39	44 db
TV 8	6.75 to 12.75	58.9	4.3	23 db	0.43	43 db
Base Band	D. C. to 6.0	0	6.1	--	0.61	--

Full Scale Volts = 2.0 volts peak to peak.

Table 3-21

VOLTAGE VALUES

E_{in}	E_{out}	Error Volts	%Full Scale Error
+1.0	1.002	0.002	0.1
0	0	0	0
-1.0	-1.000	0	0

Circuit accuracy as shown in Table 3-21 may be able to be built. In the event linearity cannot be controlled, it may be necessary to put analog functions on one data bus and PCM on another.

Analog functions will have some interference, but the PCM interference with each other will be negligible because the carriers, 140, 180, and 260 were selected to have no 2nd or 3rd order intermodulation products being within the band.

3.3 DATA BUS CONFIGURATION

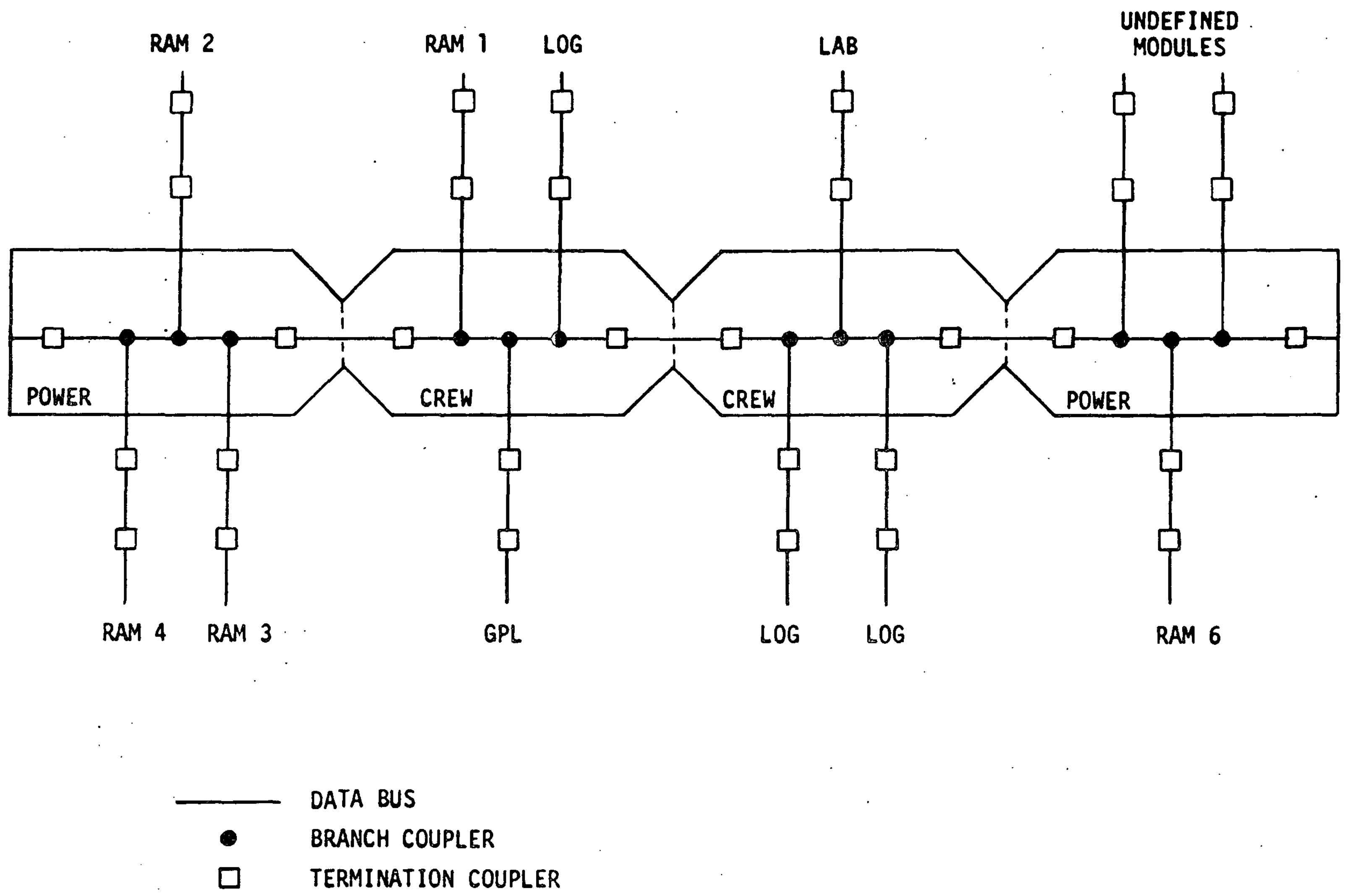
The Space Station data distribution system must be capable of handling both analog and digital data. The data bus system is described in the following paragraphs. A design specification for the system is contained in Appendix B.

3.3.1 System Description

The data distribution system consists of two FDM data buses with the capability of handling the number of channels and terminations discussed in Section 3.1. The data bus distributes data by connecting with all modules of the Growth Space Station.

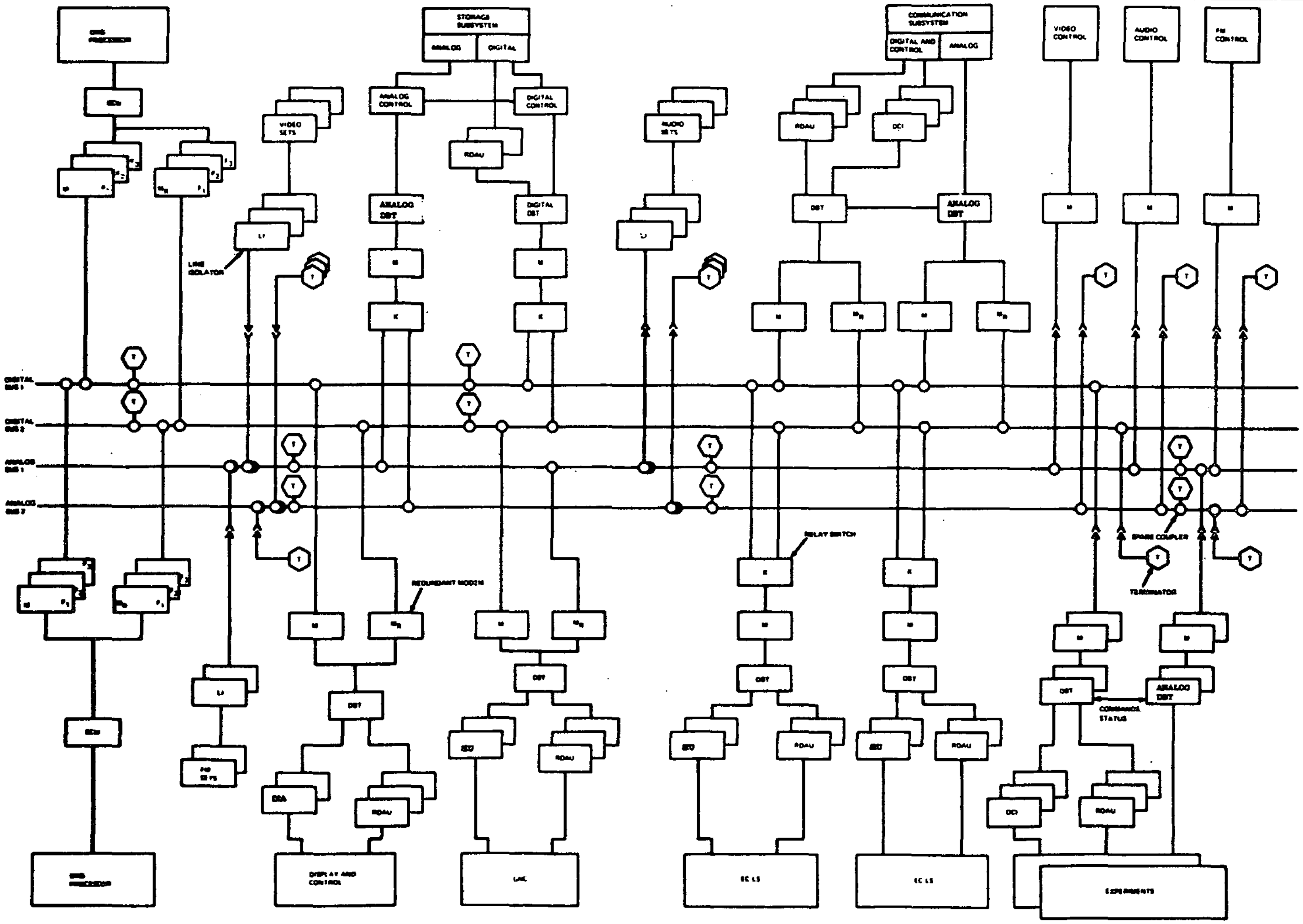
The connections to these modules are shown schematically in Figure 3-20. Each interface with the data bus is a termination. The data bus is composed of two major segments: the main line data bus and the branch line data buses. The main line data bus connects the two power modules and the two crew modules. Each digital and analog station in these modules is connected directly to the corresponding bus through three port couplers. Each analog and digital station in a branch connect to the branch data bus through three port couplers, and the branch data bus connects, in turn, to the main line data bus through another three-port coupler. The digital and video couplers are designed specifically for the respective frequency band. The audio and branch point couplers are wideband resistive networks having the outlined characteristics from DC to 900 MHz.

The total data distribution subsystem is shown in Figure 3-21. A redundant bus is shown for both the analog and digital buses, with three types of switchover. Completely redundant modems and terminals would be supplied for



NOTE: Same connection concept for both Analog and Digital Buses.

Figure 3-20. Data Buses Connection Diagram



DATA DISTRIBUTION SUBSYSTEM

Figure 3-21. Data Distribution Subsystem Design

critical functions, thereby allowing atuomated remote switchover. Less critical, but highly desirable functions would be remotely switchable through a relay switch, but this would not be an automatic switch. The least critical functions can wait until manual switching takes place by removing a connector from one bus and connecting it to the redundant bus.

3.3.1.1 Analog Data Bus

The analog data bus provides for audio and video transmission directly on an assigned FDM channel. These channel assignments are shown in Figure 3-22.

Branch-to-branch transmissions take place with no degradation of signal quality for either system.

The audio system utilizes single sideband suppressed carrier and uses sub-multiplexing techniques to pack the 48 audio channels in the spectrum shown in Figure 3-22. The video system utilizes standard CATV techniques but employs a higher safety margin by using additional 2 MHz guardbands between video channels. The couplers used to connect the analog stations to the bus are shown in Figures 3-23, 3-24, and 3-25.

3.3.1.2 Digital Data Bus

The digital data bus contains three 10 MBPS FDM channels located at 140 MHz, 210 MHz, and 300 MHz. Each channel is 25-MHz wide in order to accommodate the 10 MBPS Manchester Type II coding.

The digital data bus interfaces with special processors, display units, recorders, RDAU's, and experimental packages through the use of data bus terminals.

The digital data bus is connected to the data bus terminals through a three-port connector which connects to a modem through a power splitter. The connection schematic is shown in Figure 3-26.

(1) Public Address, Emergency Call, 300 to 3,000 Hz

(2) Telephone Carrier Reference, 4,000 Hz Sine Wave

(3) Emergency Call Tone, 5,000 Hz Sine Wave

(4) Emergency Alert Tone, 6,000 Hz Sine Wave

(5) 36 Telephone Channels, 60 KHz to 252 KHz,
SSB-Sc-AM, 300 to 3,000 Hz Audio

(6) 3 Entertainment Channels, 1.0 MHz ±75 KHz,
1.15 MHz ±75 KHz, 1.3 MHz ±75 KHz, Frequency Modulated

(7) Television Carrier Reference, 4 MHz Sine Wave

(8) to (15) Television and Video Channels, 4.75 MHz Baseband,
6 MHz Vestigial Sideband AM, Carrier Frequencies Spaced
at 8 MHz Intervals Starting with 8 MHz, 2 MHz Guard Band.

(16) Onboard Generated Test Channel, 4.75 MHz Baseband,
6 MHz Vestigial Sideband AM, Located in the Inter
of 70.75 to 76.75 MHz.

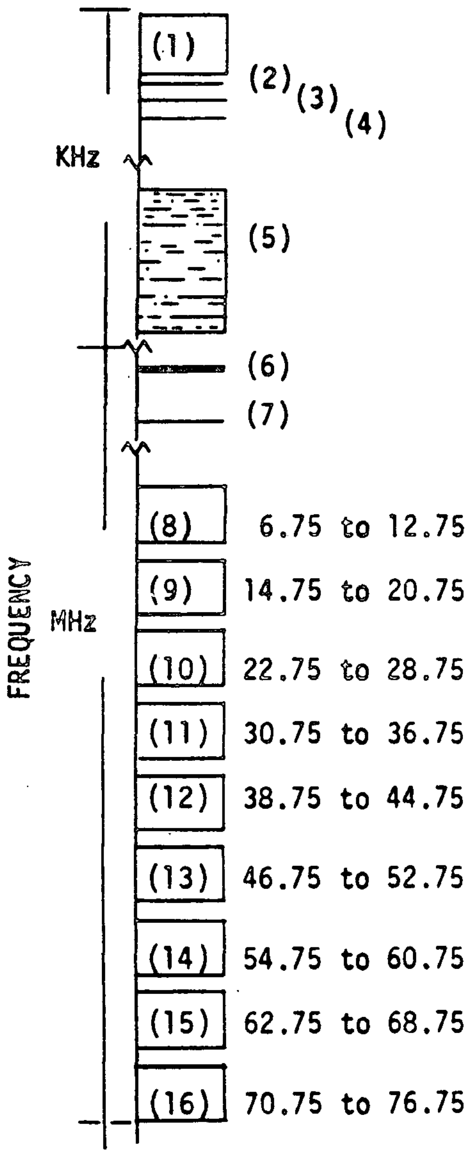
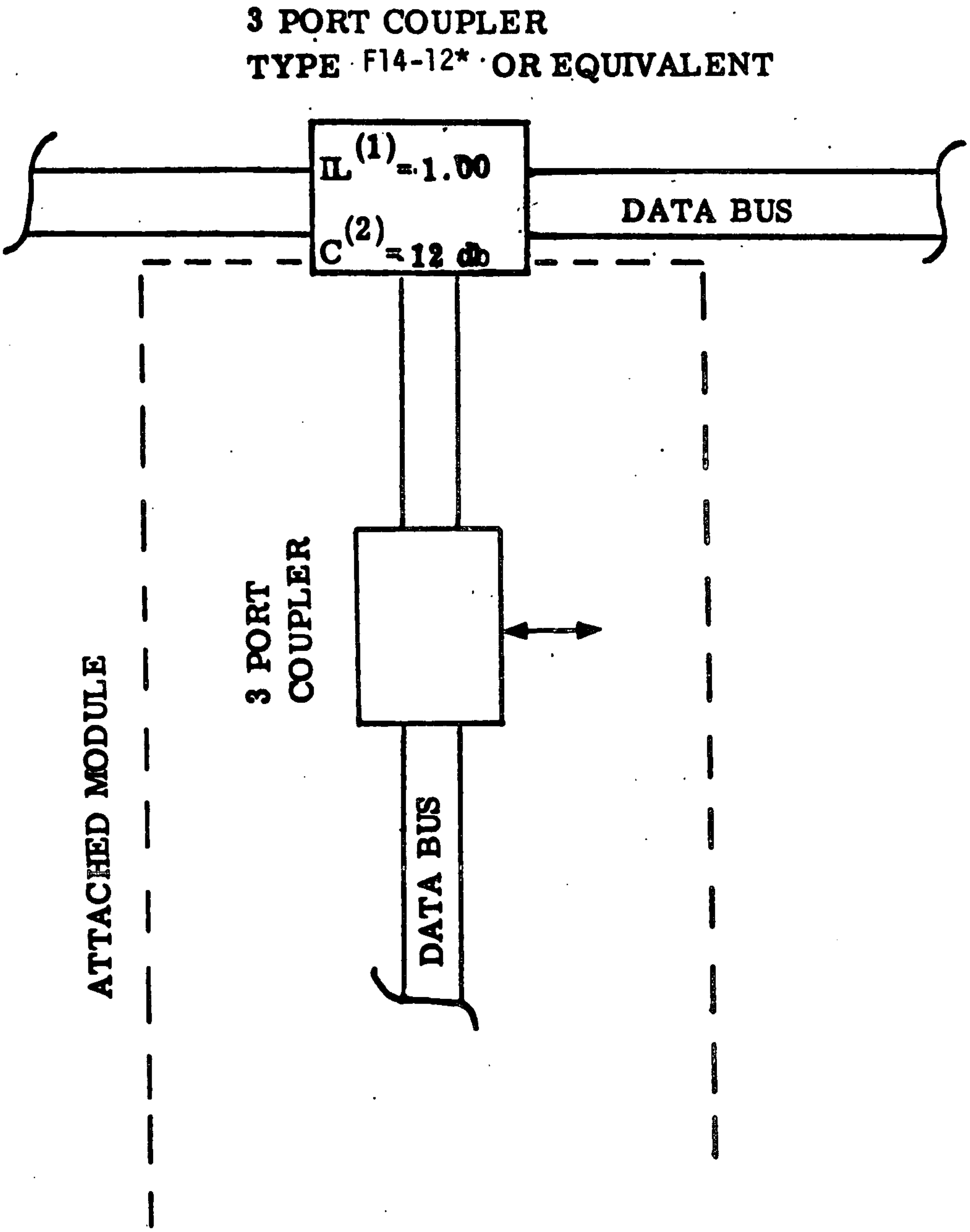


Figure 3-22. Analog Channel Allocation Chart



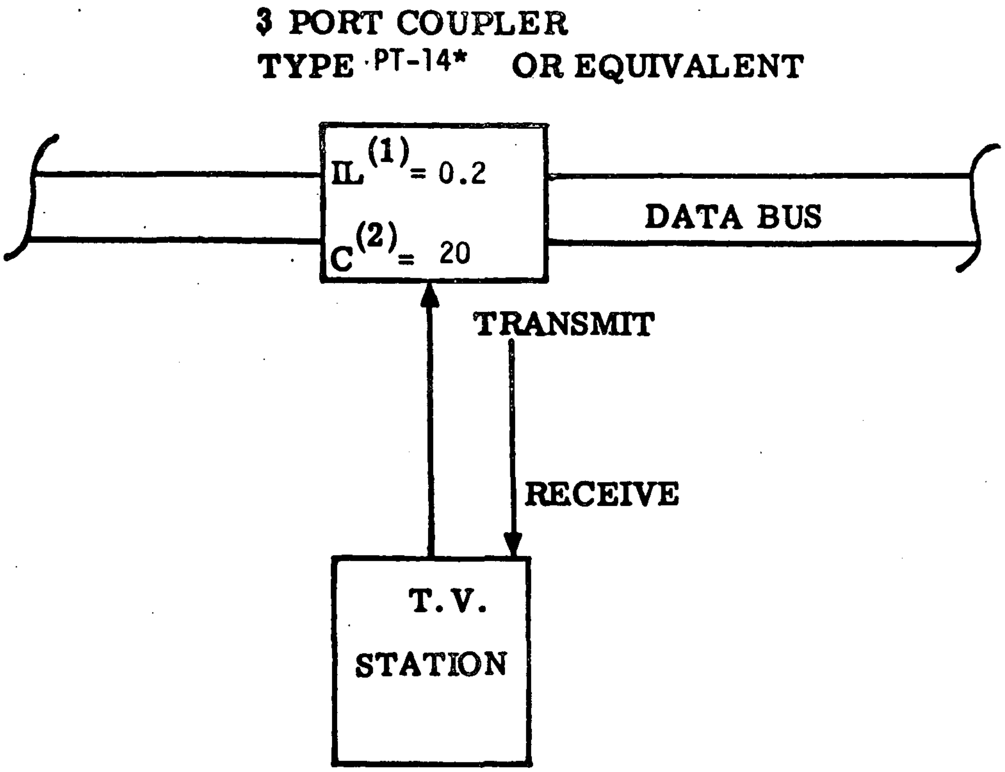
IL(1) = INSERTION LOSS BROADBAND

C(2) = COUPLING LOSS BROADBAND

* JERROLD ELECTRONICS

Figure 3-23. Branch Port Connection

5

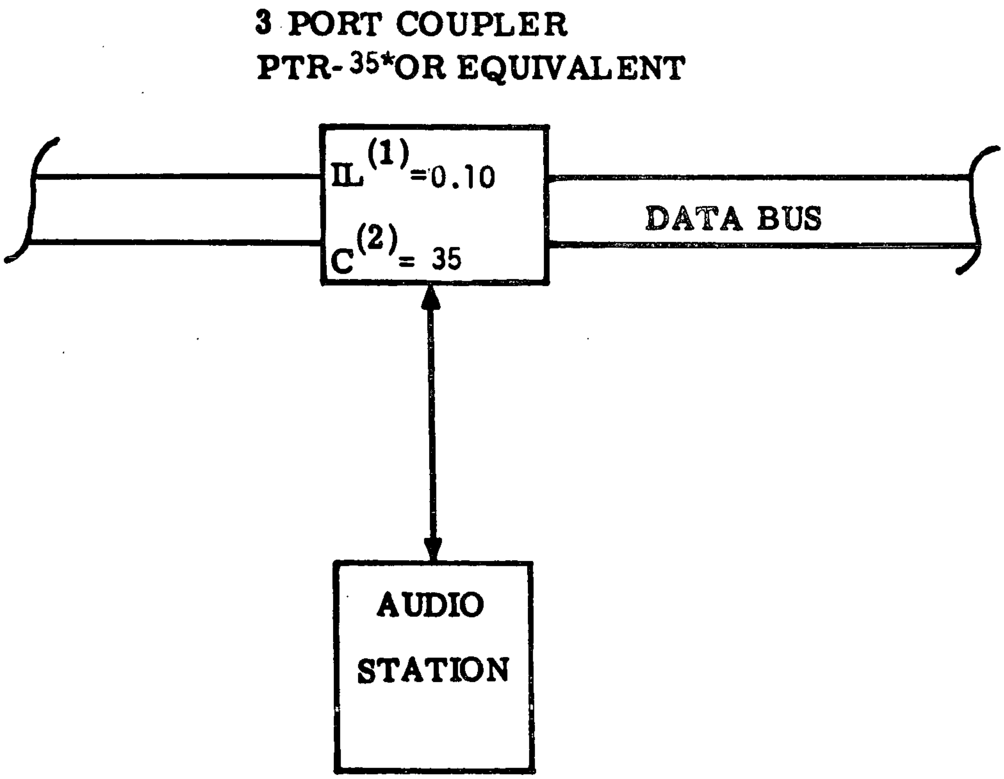


$IL^{(1)}$ = INSERTION LOSS BROADBAND

$C^{(2)}$ = COUPLING LOSS BROADBAND

* JERROLD ELECTRONICS

Figure 3-24. Video Channel Termination

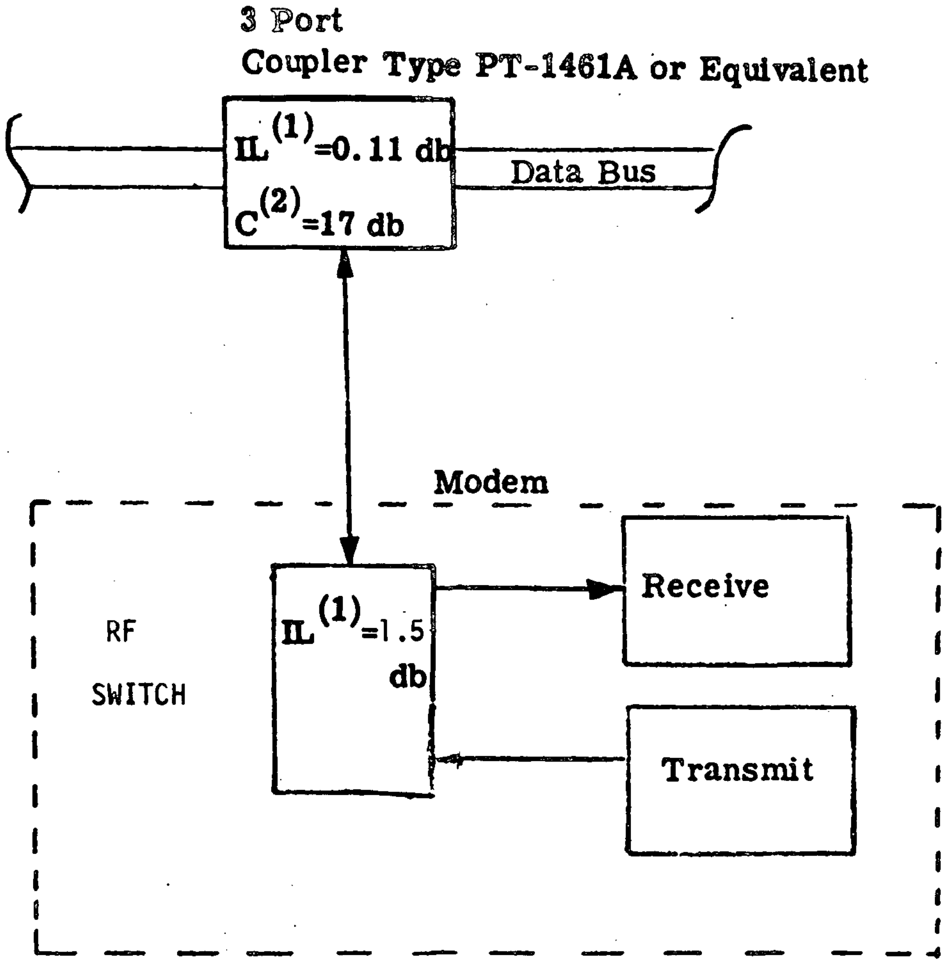


IL(1) = INSERTION LOSS BROADBAND

C(2) = COUPLING LOSS BROADBAND

* JERROLD ELECTRONICS

Figure 3-25. Audio Channel Termination



IL$^{(1)}$ = Insertion Loss At 140 MHz

Figure 3-26. Digital Channel Termination

The data bus terminal (DBT) interfaces digitally with the data bus and is capable of performing the following functions:

A. Receive, check validity, and decode commands addressed only to that DBT.

B. Store RDAU channel data for subsequent transmission.

C. Store experiment data for subsequent transmission.

D. Demodulate subcarrier.

E. Establish bit and word synchronization.

F. Load data register (terminal, line, read/write, function, channel, etc.)

G. Check parity.

H. Decode terminal address:

1. If not receipient, reset register.
2. If receipient and no parity check, update status word, transfer controller address and status word to response register; transfer status word to controller by internal clock onto data bus.
3. If receipient and parity, continue.

I. Decode line address and enable proper gate.

J. Transmit commands.

K. Decode command/data bit:

1. If data, set ready bit in status word; transfer controller address and status word to response register; transfer onto data bus.
2. If command, set command register and select proper response. Set reponse register and transfer to data bus, wait for next command or execute.

L. Indicate end of message.

Figure 3-27 is a block diagram of the data bus terminal. Each data bus terminal can receive one message at a time on one frequency channel. This particular task or message must be executed before another command can be received. Up to eight channels, such as RDAU's, experiment packages, storage devices, etc., may be connected to a data bus terminal. The channel address is decoded in the terminal, and the proper commands are routed through the input/output switching unit. The length of the command message is determined by the requirements of the receiving function. RDAU's require

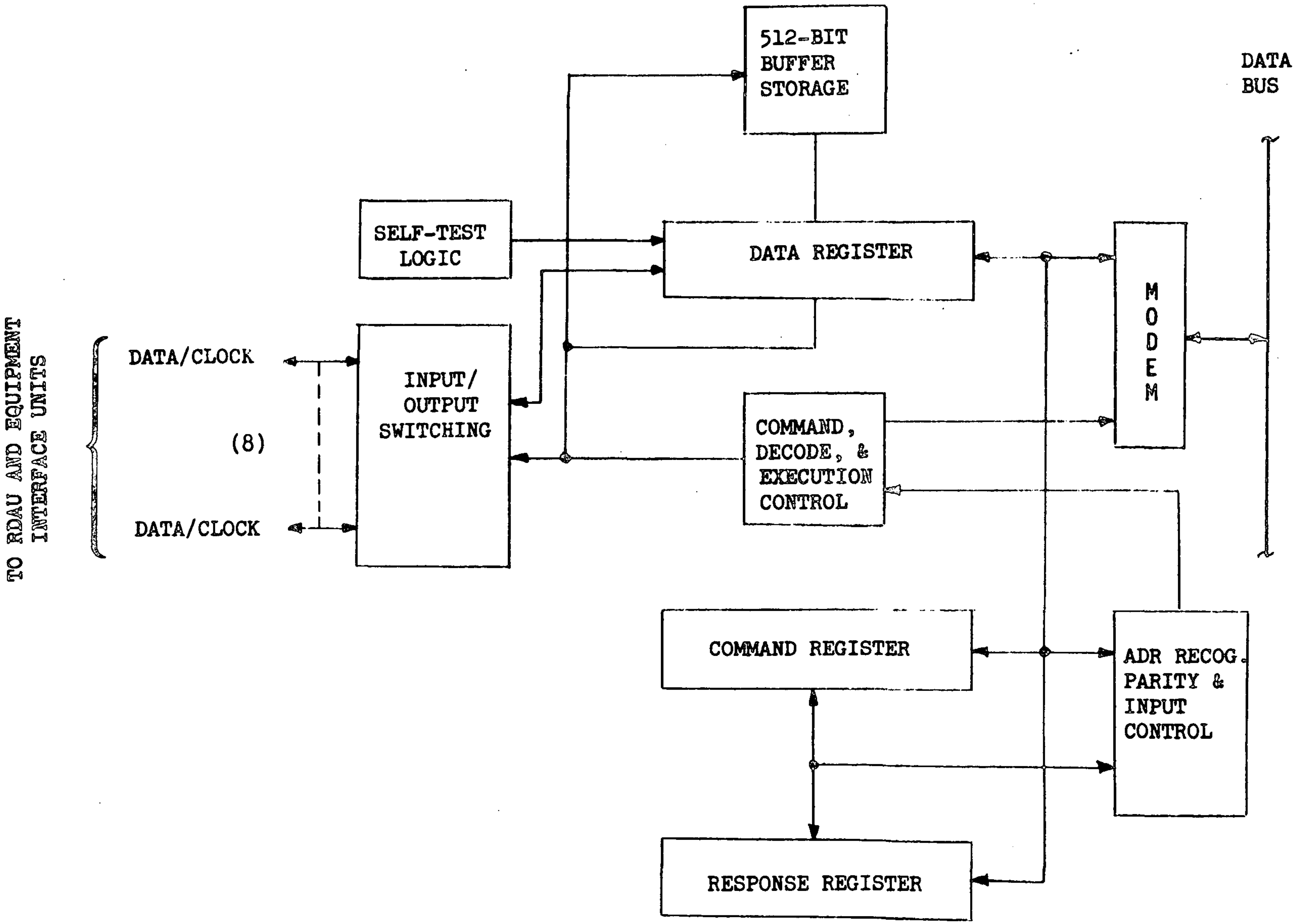


Figure 3-27. Data Bus Terminal Concept

one or two command words, while certain experiments and/or subsystems such as displays may require considerably more.

Since the data bus terminal is the interface between RDAU's, experiment packages, displays, etc., and the data bus, it must also be capable of transmitting specific commands to any or all such terminated devices through the input/output switching unit. It must also act as a transparent buffer for experimental data being inputted to the data bus. The command, decode, and execution control function provides these features in conjunction with the data register. This register acts as a transfer register between the modem and the input/output switching unit and as an input/output register for the buffer storage unit.

The RDAU must work in conjunction with the DBT, and provides the interface between sensor and the DBT. To accomplish this, the RDAU performs the following interface functions:

A. Receives command from DBT.
B. Sets differential programmable gain amplifier (DPGA) gain.
C. Changes stored limits for comparison mode of operation.
D. Provides automatic limit checking operation.
E. Scans all inputs sequentially.
F. Transmits data to DBT.
G. Sets and resets discrete command outputs.

The block diagram of the RDAU is shown in Figure 3-28 and provides the following operations:

A. Signal Conditioning -- Accomplished through preconditioning networks ahead of the analog multiplexer and a programmable gain amplifier immediately following multiplexing. The signal conditioning provides the required gain/bandwidth characteristics required for a particular application.
B. Multiplexing -- Both analog and digital multiplexing circuits are provided to accommodate up to 16 channels of digital data multiplexed in groups of eight and up to 30 analog channels. The multiplexers can handle any combination of analog channels and any combination of sets of eight digital channels.

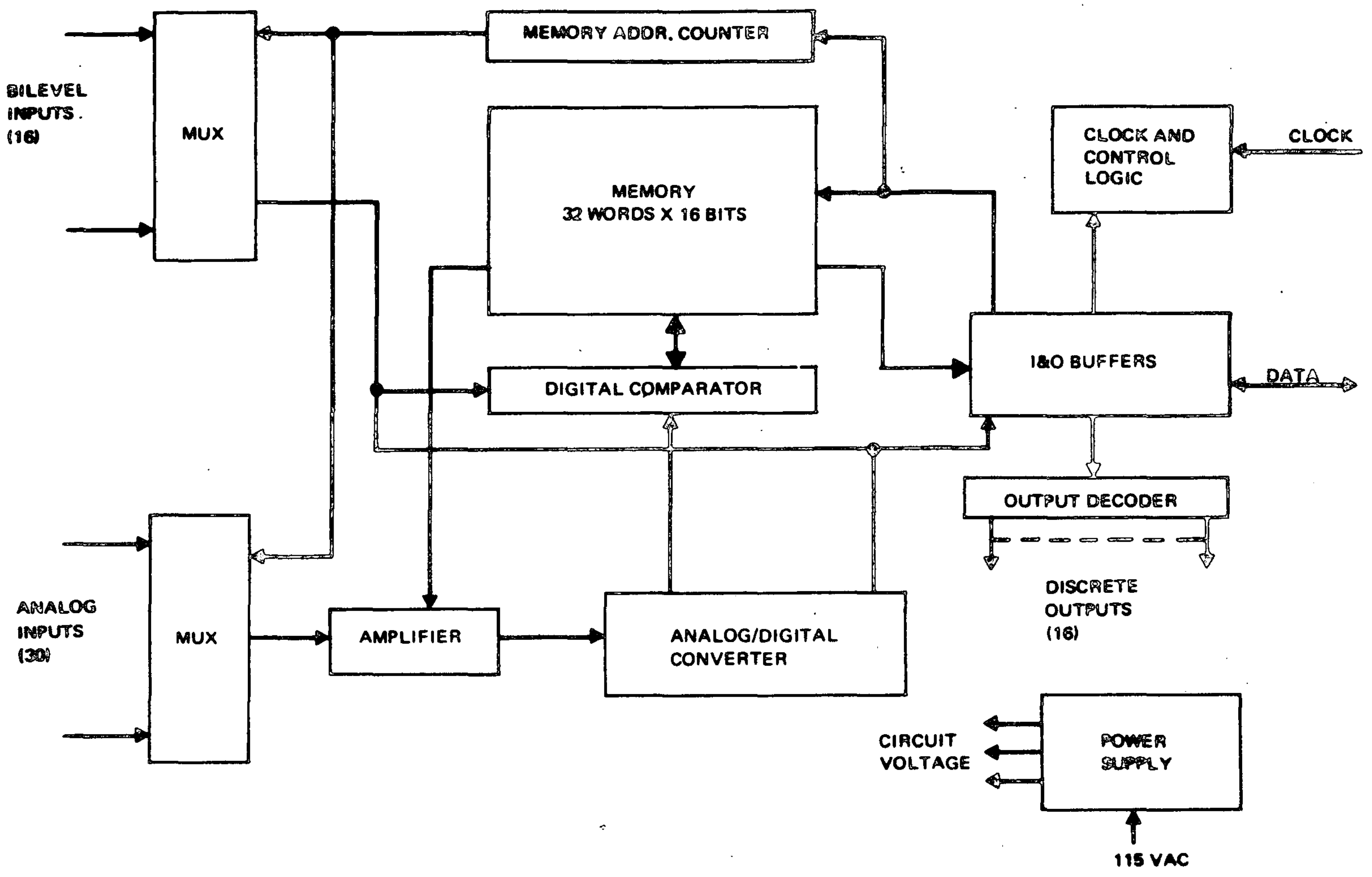


Figure 3-28. Remote Data Acquisition Unit

C. A/D Conversion -- The A/D conversion bit rate equals the clock rate in the RDAU. The A/D converter output is digitally compared with high and low limits extracted from a self-contained memory. If the measured parameter exceeds either limit, the return data is flagged with the out-of-limit channel address as an out-of-tolerance condition.

D. Digital Outputs -- The control logic allows for 16 control function outputs to provide on-off control. A separate control signal is required to switch to either the on or off state.

3.3.2 Operating Concepts

3.3.2.1 Analog Data Distribution

The analog data distribution system allocates data such as video and audio directly to a dedicated FDM channel. No other source can occupy this particular frequency channel. Any receiver capable of being tuned to that frequency can monitor all transmissions in that channel. Thus, several video monitors, such as at the commanders control station and the experiment control station, may be monitoring the same transmission, much like CATV. The control of the transmission devices may be accomplished remotely and automatically where the device controls are connected to the digital data bus or where analog controls are connected to separate FDM channels interfacing with a command decode functional unit. Most controls, such as on-off switches, could be maintained remotely through the use of the control functions of an RDAU on the digital data bus. However, it is anticipated that most of the analog system control will be accomplished manually. On-off functions could be handled remotely from the control stations, but such things as video receiver channel selectors will be controlled manually, either with a rotary- or pushbutton-type channel selector switch.

The audio system contains an emergency voice channel at baseband which preempts all other audio channels and bypasses all on-off switches. In addition, three FM entertainment channels will be available (probably on a 24-hour basis) for astronauts who are off duty. Previously recorded analog magentic tapes would be selected according to channel and time.

Telephone channels will be handled with standard telephone circuit techniques, except that no central switching unit is necessary with the 48 channels. These channels provide enough capacity to carry all required station-to-station conversations, Space Station-to-ground calls, and conference calls requiring up to eight different stations tied in simultaneously. By using the proper dialing code, a Space Station astronaut may tie directly into the telephone networks on the ground, thereby enabling a direct interface with the Earth's scientific community.

3.3.2.2 Digital Data Distribution

The digital data distribution system transfers data by the use of an asynchronous TDM/FDM data bus. Three 10-MBPS digital channels are used in a command response operational mode to handle the enormous amounts of generated data. The required data rates shown in Section 3.1 represent one high-rate experiment. During GSS operation, several of these experiments will be operational. The data-bus operational concept assumes the capability of scheduling so that no more than one of these experiments operates at the same time. If this is not the case, more 10-MBPS channel will have to be implemented to handle the immense peak load represented by two simultaneously operated high-rate experiments.

All digital operations on the data bus are controlled by the executive routine of the subsystem multiprocessor. (In the event of scheduled maintenance or emergency shutdown of this processor, bus control will be assumed by the experiment processor.) Each device connected to the data bus is controlled by a data bus terminal (DBT) which serves as the interface between the data source and the data bus. One of three data bus channels is available to each DBT by the proper modem selection and some DBT's contain up to three modems to allow different data channel selections.

Each terminal is commanded by the control processor at preprogrammed intervals. These intervals may vary depending on the application and are adjustable by making slight modifications to the executive program. The most common interrogation intervals are 0.01, 0.1, 0.5, 1 and 10 seconds. As an example, assume an astronaut keys a command into a control keyboard.

This command is stored until retrieved by the processor some 0.01 seconds later. The command may actually represent a series of commands. This may be interpreted by the processor and a command chain sent to the addressed device. Data will be retreived from that device, operated on by the processor as required and probably displayed. The total reaction time should be less than one-half second.

Data bus terminals interfacing with subsystem devices such as RDAU's will be commanded at regular intervals, depending on the measurement criticality.

The control of the DBT's and interfacing units is accomplished through the use of special control words, consisting of "A" and "B" words as shown in Figure 3-29. The status of the terminal is contained in the "C" word.

The processor sends instructions to a DBT by selecting the proper address for the "A" and "B" word, sending a synchronization word eighteen bits long, followed by the "A" word. If the command chain contains data, this is incorporated into the data word formats and sent directly following the "A" word transmission. Up to 32 data words may be transmitted per message. The transmission is ended with the "B" word.

If the DBT is operational and proper addressing occurred, the DBT responds with an "A" word and a "C" word. The "A" word allows the processor to validate proper terminal operation. The "C" word contains terminal status information pertaining to events since the previous command reception. After a suitable delay to allow for reset in the event of erroneous operation, the command is carried out by the DBT. The timing of these transactions is illustrated in Figures 3-30 and 3-31. Figure 3-30 illustrates the DBT terminal operation when data or commands are sent to a remote device, such as an RDAU. Figure 3-31 illustrates two types of data retrieval. The first sequence assumes that data are ready for transmission. The special request service sequence assumes that data are not immediately available and several command transactions are required to retrieve the data.

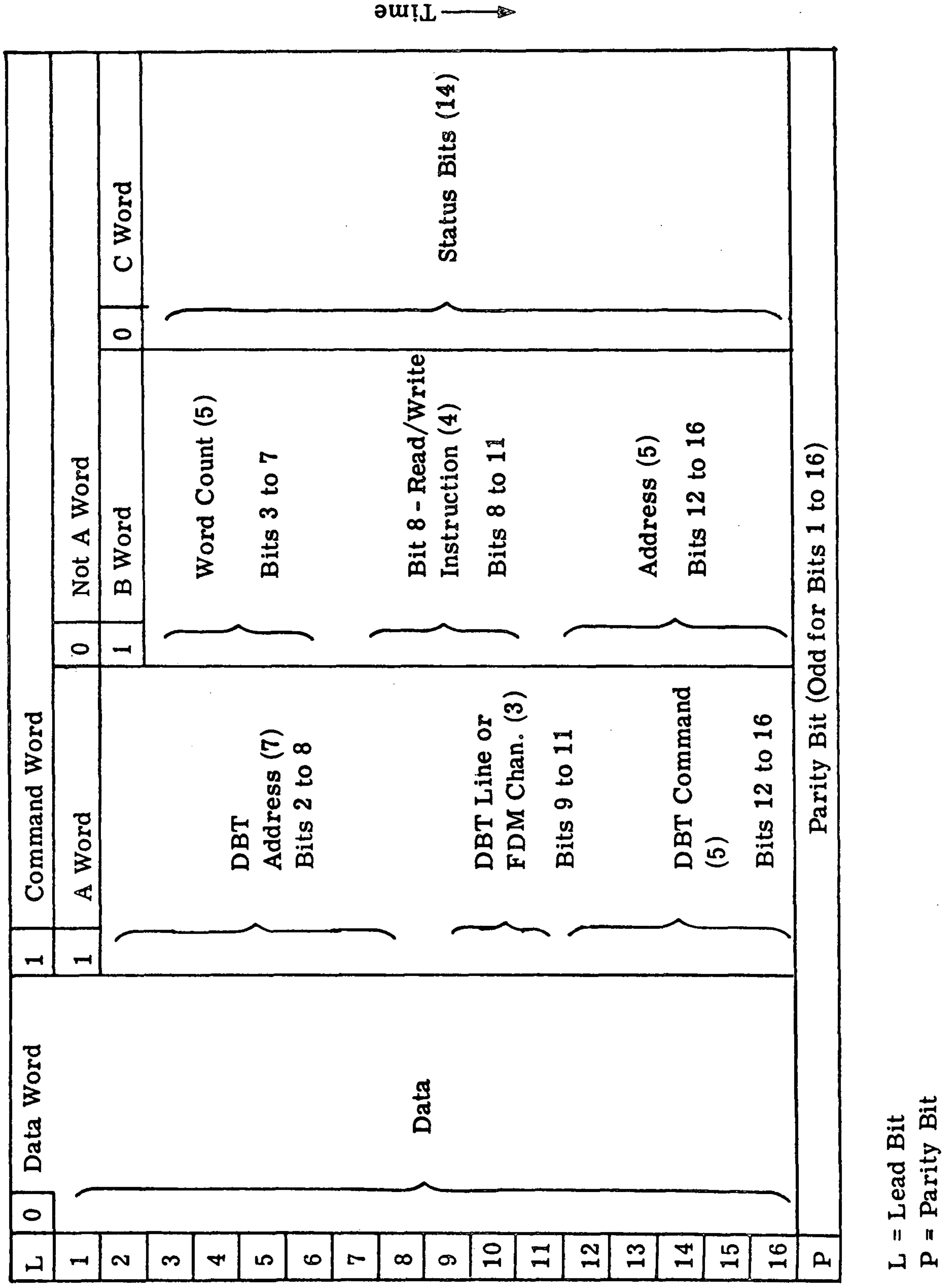


Figure 3-29. Data Bus Word Formats

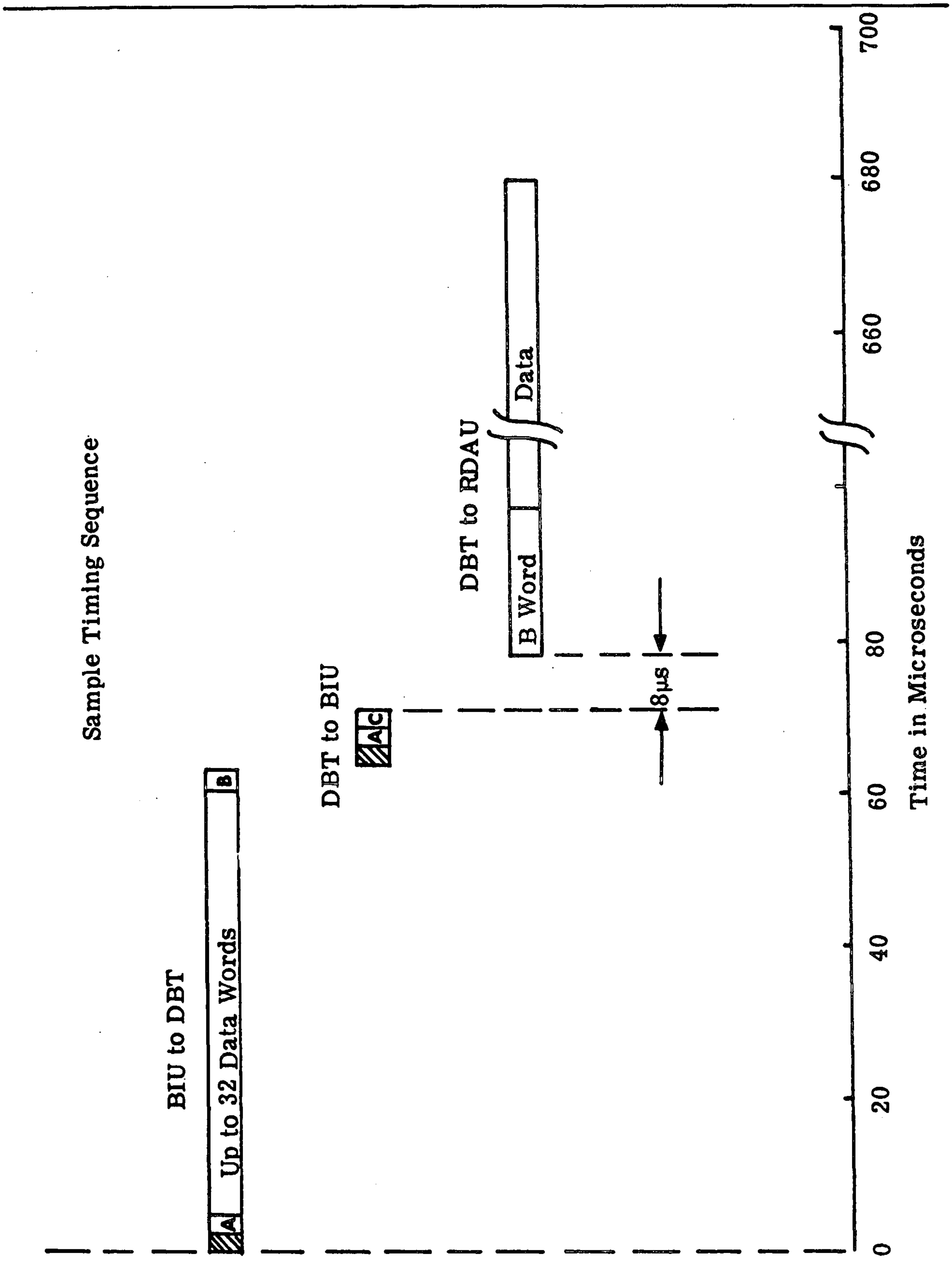


Figure 3-30. Write Sequence Timing

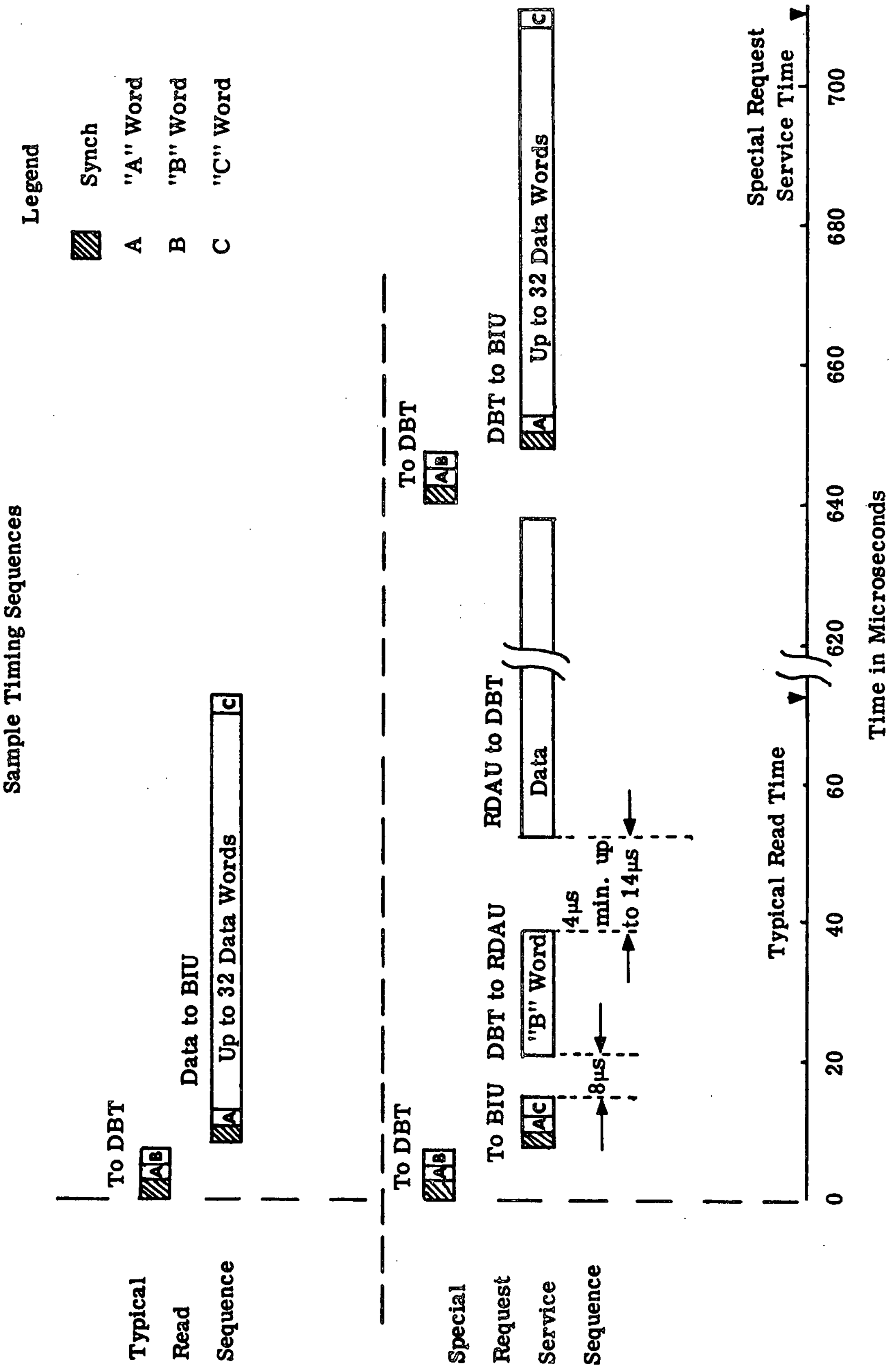


Figure 3-31. Read Sequence Timing

However, the total elapsed time is less than 750 microseconds. The data bus channel is active in this transfer for only 80 microseconds and would be used for other DBT operations while waiting for data retrieval.

The normal mode of operation involving subsystem or RDAU operation is a multiple-operation command. Most data can be called for in some time sequence and in advance of when required. Consequently, the processor will issue commands whereby the DBT transmits previously retrieved data. After data transmission to the bus, the DBT retrieves additional data either from a different I/O channel (implying a different RDAU or subsystem), a different channel of the same I/O channel, or additional data from the same channel of the same I/O channel. Meanwhile, data-bus operations are maintained by commanding other DBT's in the same manner. The usual DBT/ RDAU or subsystem interface operates at a one-MBPS speed. The 10-to-1 transmission speedup and response time requirements allow multiple DBT terminations (approximately 40) per channel. In the event that some subsystems or experiments require greater than 5-MBPS interface speeds, a special DBT would be utilized since this requirement involves only a few units. (The standard DBT can handle up to 5 MBPS I/O channel rates, provided an external clock is provided.) These special DBTS would contain higher speed logic on the I/O channel interface and could incorporate a larger buffer to cut down overhead rates where the input rates exceed 7.25 MBPS. Where enormous speeds (on the order of 50 MBPS) are required, multiple channels will have to be used. The data would have to be formatted to a parallel/serial wavetrain and inputted to several DBTs operating on different channels. Skew problems from different channel delays would have to be handled by the processor.

The data bus terminal is an important segment of the digital data bus. It interfaces between a 10-MBPS Biphase Manchester Type II coded data bus channel and any one of eight one-MBPS Bipolar NRZ coded I/O channels. The DBT interfaces with the data bus through a modem operating on a single frequency data bus channel. Logic and addressing are included to permit expansion to three 10-MBPS, FDM channels. Each DBT responds to a uniquely assigned hardwire programmable address.

The eight DBT I/O channels transfer data and commands sequentially to and from peripheral devices such as RDAU's or experiment subsystems at a one-MBPS rate. The DBT exercises master control over the peripheral devices. Data transfer between the bus and the DBT occurs only by command from the bus at a 10-MBPS rate. The DBT has the capability to "read" data from one of eight channels. It also has the capability to "write" commands or data to any specific channel. The DBT recognizes a special command for device-to-device transfers, whereby one DBT communicates with another (for example, an experiment-to-tape communication).

The DBT contains the major functional blocks and registers shown in Figure 3-32. Performance of the DBT is as indicated by the flow diagram of Figure 3-33 in accordance with the format and functional descriptions as contained in the following paragraphs.

Each DBT, after recognizing an "A" word following a synch burst, performs the following functions:

A. Address Decoding — Each DBT contains its own unique seven-bit address code programmable by changing a plug-in unit. The "A" word of any command contains the address code for a particular DBT in bits 2 to 8. Every DBT on any given FDM channel shall examine and "A" word following a synch burst for its particular address code. Upon recognition of this address code, the DBT accepts subsequent words for command (function) decoding and data handling. If the address code in invalid, the DBT discontinues logic decoding until the arrival of another synch burst.

B. Function Decoding — Every "A" word also contains a command or function code bits 12 to 16 used to provide an alerted DBT to perform a specific function. Functions are decoded after a valid address has been received.

C. Channel Address and Function Decoding — Each DBT controls eight I/O channels addressed by bits 9 to 11 in the "A" word. Specific I/O devices are then supplied data and/or commands through the DBT. I/O device subchannels and commands are contained within the "B" word.

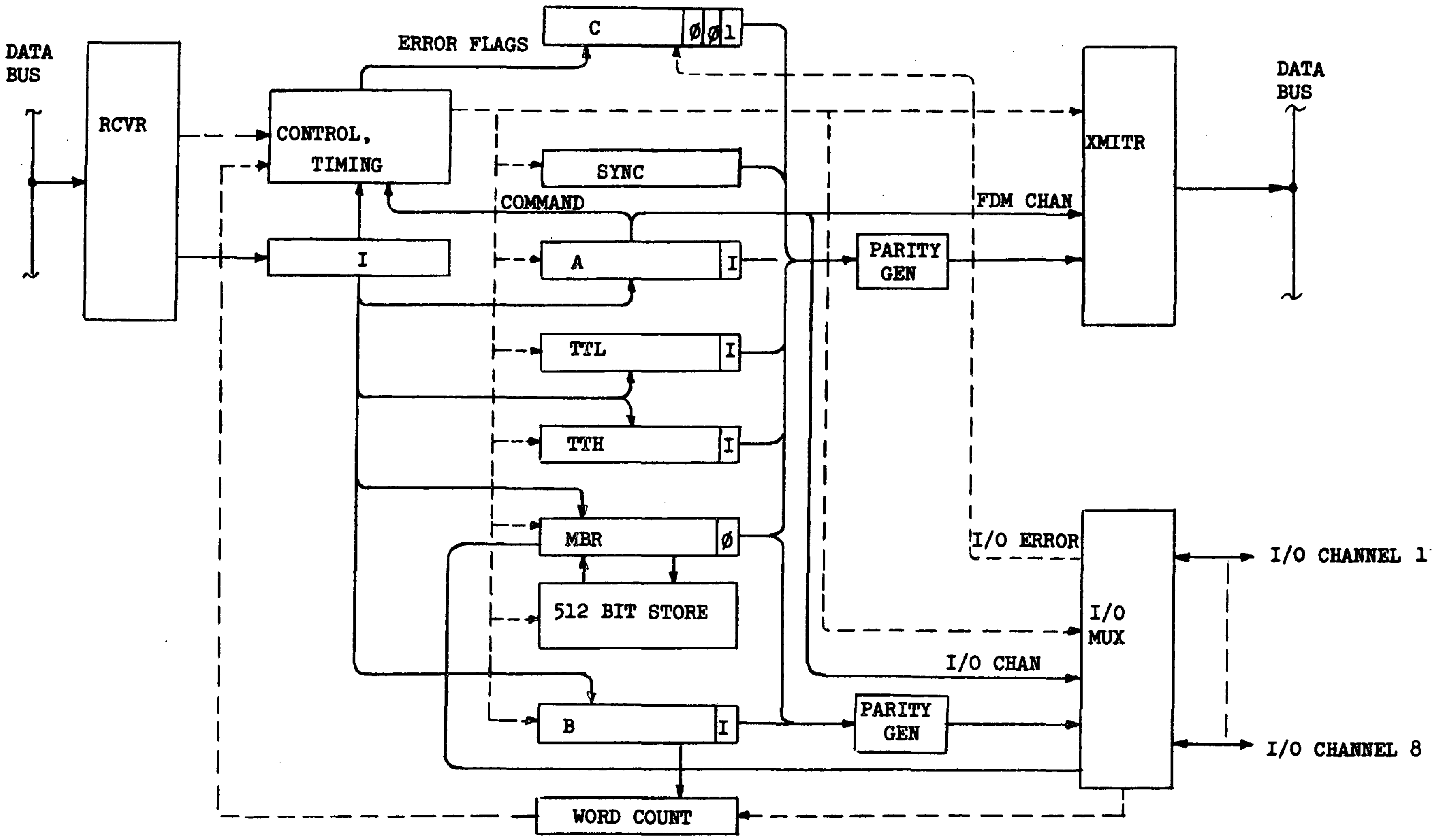


Figure 3-32. Data Bus Terminal Block Diagram

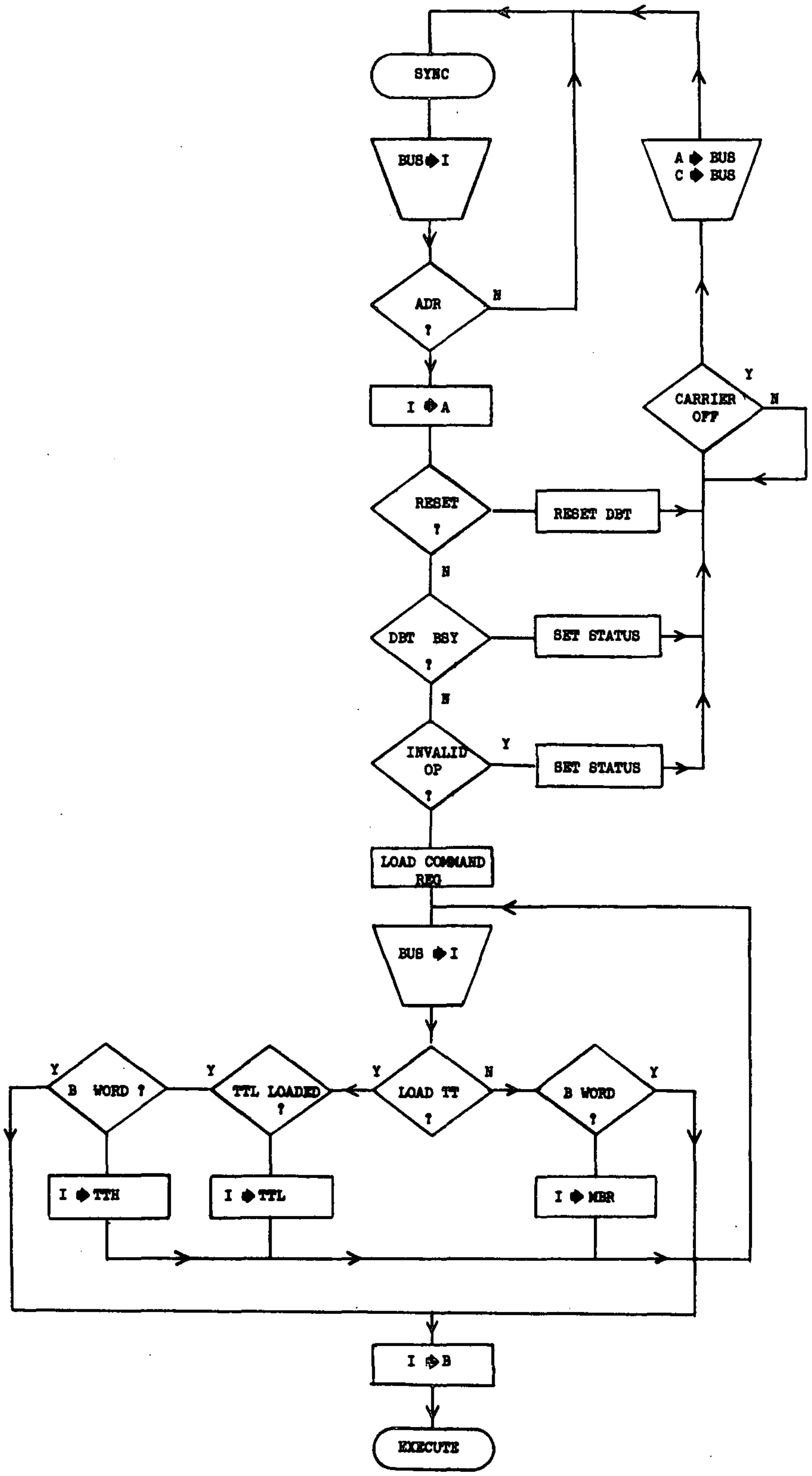


Figure 3-33. Data Bus Terminal Flow Diagram (page 1 of 3)

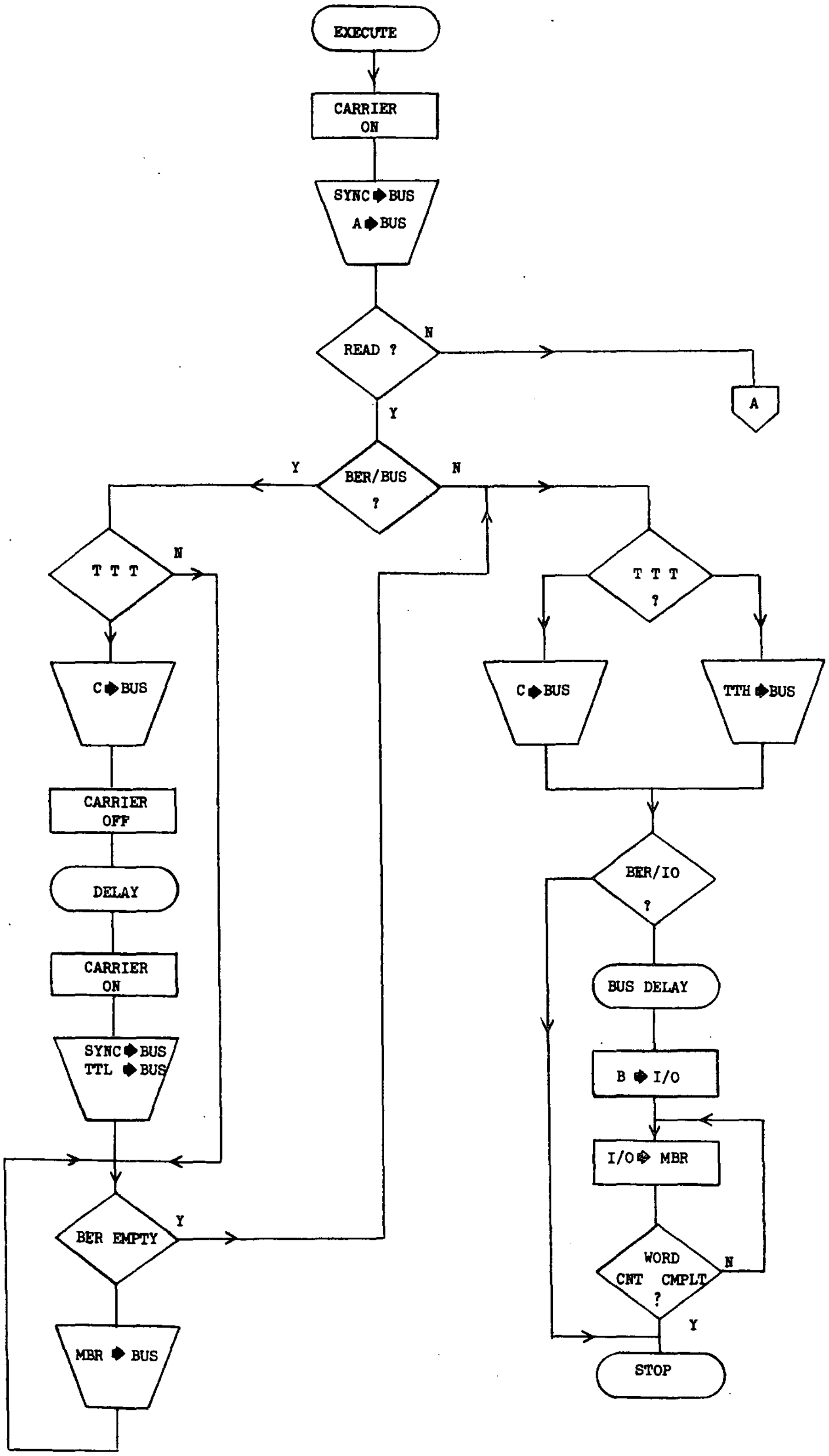


Figure 3-33. Data Bus Terminal Flow Diagram (page 2 of 3)

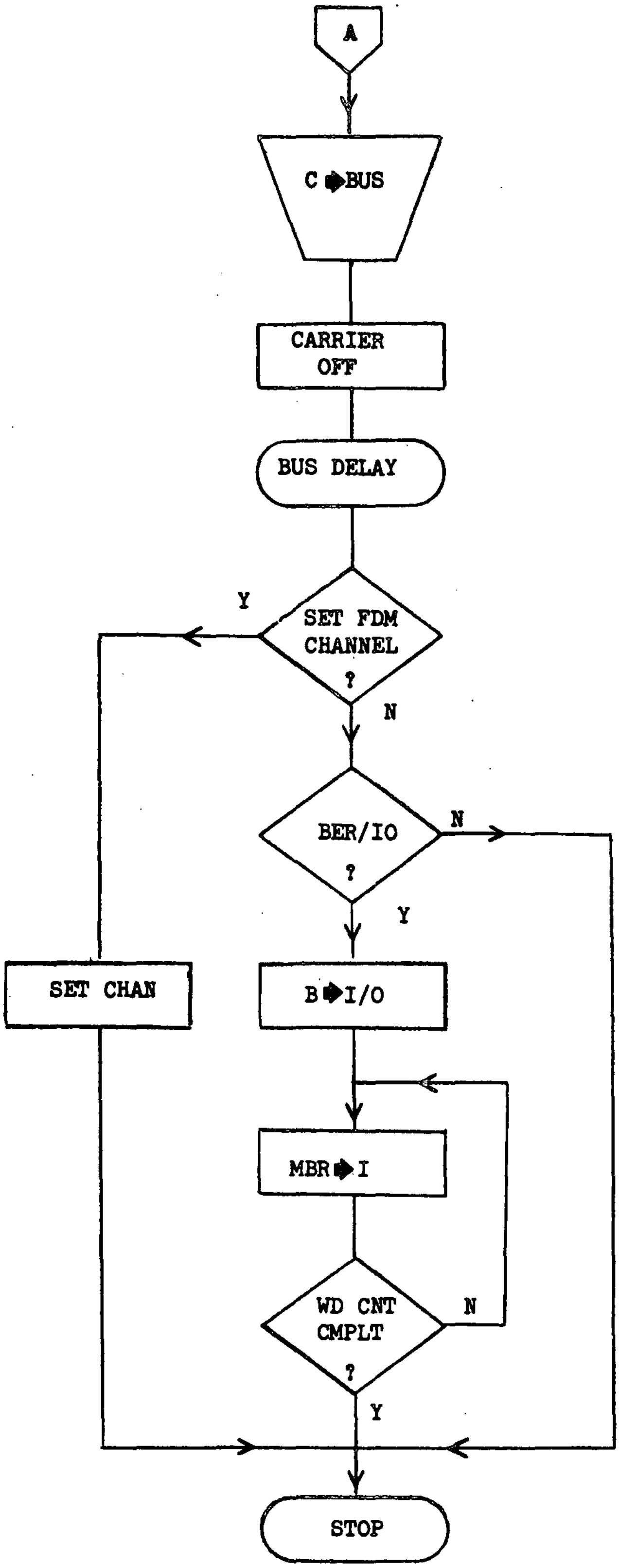


Figure 3-33. Data Bus Terminal Flow Diagram (page 3 of 3)

D. DBT Programming – The following is the DBT command structure and a description of the valid commands and their operation.

The DBT command field consists of five bits, each with its own particular significance as illustrated in Figure 3-34.

The valid DBT operations are described below.

A.	11111	RESET DBT
B.	10000	LOAD TTT REG.
C.	10001	SET FDM CHANNEL
D.	00000	NO OP
E.	00101	READ I/O TO BUFFER
F.	0X110	READ BUFFER TO BUS
G.	0X111	READ BUFFER TO BUS AND I/O TO BUFFER*
H.	00100	NO OP
I.	00001	WRITE BUFFER TO I/O
J.	00010	WRITE BUS TO BUFFER
K.	00011	WRITE BUS TO BUFFER AND BUFFER TO I/O

The DBT operations associated with each command are as follows:

A. RESET DBT – This command has highest priority and shall cause any operation in process to stop and resets the TTH, TTL, and C registers.

B. LOAD TTT REG. – In this command, the two data words from the bus shall be loaded into the TTL and TTH registers. These registers shall be cleared by loading all 0's on receipt of the reset command.

C. SET FDM CHANNEL – This command allows a DBT to be reassigned to a different FDM channel. The DBT response shall be an "A" word and a "C" word transmitted on the previously assigned channel frequency. Following the response, the DBT will switch to the new frequency.

D. NO OP – This command causes the "A" and "B" registers to be loaded. The DBT responds with an "A" and a "C" word but no other operation is performed.

*A "1" in position X will result in the operations being performed as a TT transfer.

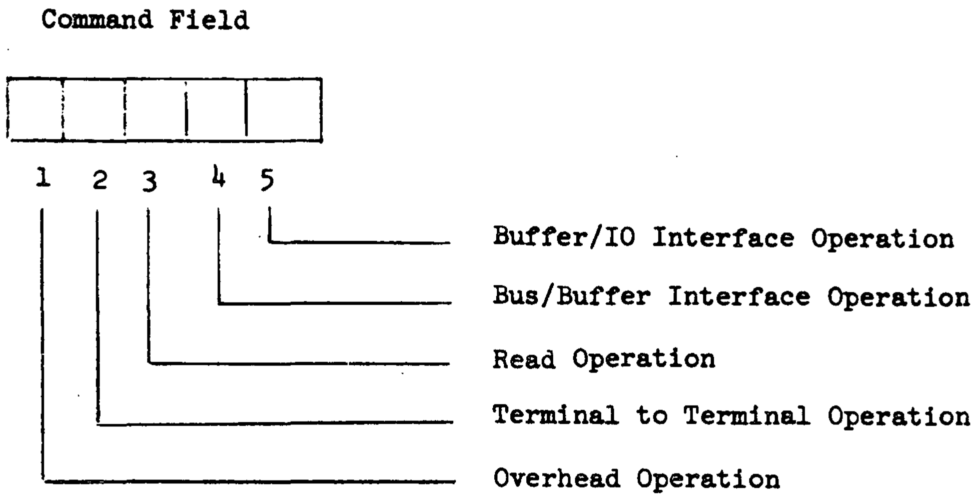

Command Field
1
2
3
4
5
Buffer/IO Interface Operation
Bus/Buffer Interface Operation
Read Operation
Terminal to Terminal Operation
Overhead Operation

Figure 3-34. Command Field

E. READ I/O TO BUFFER – This command is received as an "A" and a "B" word from the bus. Immediately following address and command decoding a "response" is sent over the bus, consisting of the original commanded "A" word followed by a "C" word. Once the command has been received and the response sent, execution of the command begins after an 8-second delay. The I/O channel address is extracted from the "A" word. The "B" word is transmitted over the I/O channel and response data is received back. The number of received data words corresponds to the number specified in the count field, bits 2 to 7, of the "B" word. The data are loaded into buffer storage.

F. READ BUFFER TO BUS – In order to access the buffer, this command may be sent which causes the contents of the buffer storage to be transmitted onto the data bus as data words. These words appear, in time, between the response "A" word and the "C" word. No further action following transmission of the status word takes place in the DBT.

G. READ BUFFER TO BUS AND I/O TO BUFFER – Operation of this command is a sequence of commands 5 and 6. The buffer is transmitted to the bus per command 6, but following the "C" word transmission, the I/O channel is read into the buffer per command 5.

H. NO OP – Identical to command 4.

I. WRITE BUFFER TO I/O – Thiscommand causes the contents of the buffer to be transferred to the specified I/O channel. The DBT responds to the bus with an "A" and a "C" word, waits for eight microseconds, and transfers the "B" word to the I/O channel, followed by the buffer contents. The number of data words transferred to the I/O channel is specified by the count field of the "B" word.

J. WRITE BUS TO BUFFER – This command causes data words from the bus to be written into the buffer. Presence of the "B" word signifies the end of data. The DBT will respond with an "A" and a "C" word after receipt of the "B" word.

K. WRITE BUS TO BUFFER AND BUFFER TO I/O – This command is a combination of commands 10 and 9, where the buffer is filled per command 10 then transferred to the specified I/O channel per command 9.

L. TERMINAL TO TERMINAL TRANSFER – Terminal to terminal transfer is a special case of the previous "read" commands. Identical operations to those described occur except that the DBT will make two transmissions where the contents of the TT registers are used to preface the second DBT response. The TT registers contain the address of another DBT and some "write" command, as an "A′" and "B′" word. The first response is a normal synch "A" and "C" word transmission. The second response occurs after a nominal delay and consists of a synch, "A′" word, up to 32 data words, and a "B′" word. The second DBT recognizes its address and proceeds to act on the following data as if it had been transmitted by the bus controller.

The DBT provides eight one-megabit per second serial digital interfaces for communication with RDAU's or other devices. Each interface is made through three twisted, shielded wire pair cables. One cable contains a continuous one-MBPS clock output which also provides a message start signal from the DBT. A second cable provides the DBT serial Bipolar NRZ formated data output. The third cable is a serial bipolar NRZ formated data input. The DBT, when interfacing with an RDAU or other device, is capable of providing a dc isolated, transformer-coupled interface.

3.3.3 System Characteristics

The Space Station data distribution system consists of two coaxial cable data buses to handle approximately 55 FDM channels and 200 terminations. The buses are separated into analog and digital data buses.

There are 48 audio channels with one emergency channel at baseband. Each audio channel has the characteristics shown in Table 3-22. The channels are submultiplexed with single sideband suppressed carrier techniques into the frequency spectrum of 60 KHz to 252 KHz. A minimum single-to-noise ratio of 15 db insures 95 percent intelligibility.

Table 3-22

AUDIO SYSTEM PERFORMANCE CHARACTERISTICS

Receiver

Bandwidth: 3 KHz

Noise figure: 30 db

Dynamic range: 50 db

Sensitivity: Minus 105 dbm with the specified bandwidth and noise figure at 20° C ambient temperature

Transmitter

Bandwidth: 3 KHz

Modulation: Single-sideband suppressed carrier

Unmodulated Carrier

Power output: 25 dbm

There are nine video channels with the characteristics shown in Table 3-23. There are present plans for 19 TV cameras and 10 monitors, of which 6 are portable. Each channel will result in a 39-db minimum detected signal-to-noise level, assuring very fine picture quality. An exception to this occurs when a video transmission originating in an attached module connected by a branch port coupler is received in another attached module connected as a branch. System viewing will be degraded somewhat in this situation and should not be planned except for emergency or short term noncritical operation.

The nine video channels are placed in the frequency spectrum from 6.75 to 76.75 MHz and are either remotely or manually controlled on or off. The receivers are capable of switching to any of the nine channels. However, this control will be manual. The transmitters or cameras are assigned to a specific channel. If a channel must be shared by more than one camera, it is accomplished on a time-division basis according to previously determined schedules.

Table 3-23

VIDEO SYSTEM PERFORMANCE CHARACTERISTICS

Receiver

Bandwidth: 4.75 MHz (6 MHz vestigal)

Noise figure: 10 db

Dynamic range: 40 db

Sensitivity— Minus 98 dbm with the specified bandwidth and noise figure at 20° C ambient temperature

Adjacent channel rejection: 70 db minimum

Transmitter

Unmodulated carrier: 30 dbm

Power output:

Modulation: Vestigal sideband

Bandwidth: 4.75 MHz (6 MHz vestigal)

Each digital data bus channel operates at a total speed of 10 MBPS. With overhead in the form of synchronization, command words, control bits, and circuit and propagation delays, this corresponds to a true data rate on the order of 7.25 MBPS. The characteristics of each digital channel are shown in Table 3-24. Each channel is capable of interfacing with 64 digital data bus terminals in addition to the data bus controller, but only approximately 30 terminations per channel are anticipated.

3.4 BREADBOARD SUBSYSTEM DESCRIPTION

3.4.1 Breadboard Design Philosophy

The breadboard subsystem design philosophy follows the Space Station design. The information management subsystem breadboard design requirements have been derived directly from Space Station requirements, and the only deviations are found in actual blackbox hardware implementation.

Table 3-24

DIGITAL SYSTEM PERFORMANCE CHARACTERISTICS

Receiver

Data rate: 10 MBPS per channel

Data coding: Manchester Type II biphase code format

RF bandwidth: 25 MHz ± 0.5 MHz per channel

Modulation: Double-sideband amplitude modulation

Noise figure: 8 db

Sensitivity: Minus 92 dbm with the specified bandwidth and noise figure at 20° C ambient temperature

Response delay: 2.3 microseconds maximum

Center frequency: Channel 1—140 MHz
Channel 2—210 MHz
Channel 3—280 MHz

Signal-to-noise ratio: 22 db minimum

Bit error rate: 10^{-7} maximum

Transmitter

Data rate: 10 MBPS per channel

Data coding: Manchester Type II biphase coding format

Rf bandwidth: 25 MHz ± 0.1 MHz

Modulation: Double-sideband amplitude modulation

Unmodulated carrier: 20 dbm

Power output:

Turn-on delay: 400 nanoseconds maximum

3.4.2 Modem

The modem designed for the breadboard differs from the final Space Station modem design. Only the design philosophy has been followed. Off-the-shelf amplifiers and filters have been ordered and interfaces have been designed to fit the parts together into an operational system. In a design for Space Station, integrated circuitry will be utilized. Individual units such as separately packaged filters are being purchased and mounted in a large 19-inch rack-mounted unit for the breadboard in order to prove design concepts.

Circuit design concepts are not involved, and only the major functional block diagram concept will be proven by the breadboard model. This block diagram is shown in Figure 3-35.

3.4.3 Data Bus Terminal

The breadboard data bus terminal design differs from the Space Station design in two areas: packaging concept and I/O channel speeds. In the Space Station design, the packaging will be minimized for weight and volume and will be flight-type hardware. The breadboard design is packaged for a 19-inch rack mounting.

The I/O channel speeds for the Space Station will primarily be one megabit per second with the capability of handling input speeds to five megabits per second where an external clock is supplied with the higher-speed data. The design for the breadboard incorporates an I/O channel speed of one megabit per second. No provision is made for any other data rate input.

3.4.4 Breadboard Specification

Detailed design specifications for the breadboard data bus terminal, modem, and remote data acquisition unit are contained in Appendix C.

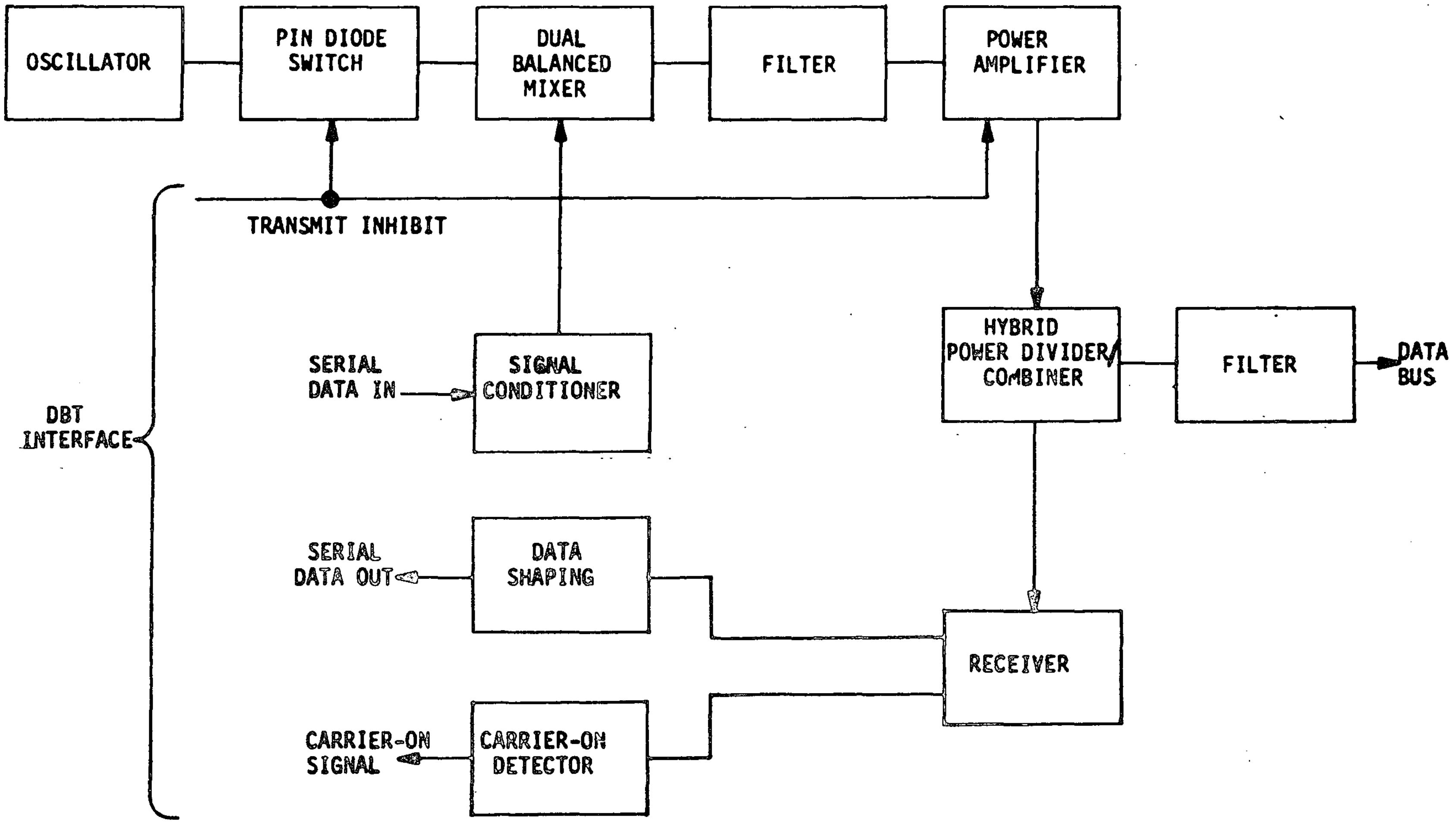


Figure 3-35. Modem Block Diagram

Section 4
DISPLAYS AND CONTROLS

Displays and controls (D&C) provide the crew with monitoring and control capability over the Modular Space Station, its subsystems, and the experiment programs. Through the D&C, a man-machine interface is achieved whereby the crew receives information from which decisions can be made and control action taken for subsystem management, vehicle maneuver supervision, energy and consumable management, checkout and fault isolation, and experiment program direction.

As a result of studies and analysis conducted during the course of the 33-ft Space Station and Modular Space Station studies, a display and control concept has evolved, which in conjunction with the data management system data bus and data processing capabilities, provides a high degree of utility with nominal expenditure of the Space Station's weight, volume, and power resources. This is achieved through the use of multipurpose computer-controlled displays and multifunction input devices to replace many of the more conventional devices such as indicator lights and dedicated control switches. This approach possesses an inherent flexibility that makes possible the efficient accommodation of a wide range of functions as well as changes and growth in the Space Station throughout its assumed long useful life.

One of the major features of the DMS/D&C design is the ability to relieve the Space Station crew from many of the routine monitoring and control functions, thus allowing a greater portion of their time to be devoted to experiment operations and other tasks. This is achieved through a high degree of automation and special display features, such as computer-controlled voice message capability that allows the crew to be alerted and notified of circumstances or events that require their attention, the caution and warning subsystem, and the onboard checkout subsystem, thus eliminating the requirement for constant manning of the control station.

Among the objectives of the IMS special emphasis task is the review and updating of Modular Space Station display and control requirements and methods of implementation, and the construction of a breadboard D&C console to allow demonstration, development, and evaluation of selected concepts.

4.1 MODULAR SPACE STATION DISPLAYS AND CONTROLS

The display and control configuration for Modular Space Station is described in document SE-04, Preliminary Subsystem Design, and is briefly summarized here. The configuration consists of a primary command and control center and an experiment and secondary command and control center.

The primary command and control center is the central command post for the MSS and is the focus of all Space Station mission control activity. This station provides monitoring and control capability of all subsystem housekeeping activities, mission planning, personnel activity scheduling; plays a central role during all rendezvous and docking phases with other spacecraft and free-flying modules; and directs communications for command and data transfer to/from other spacecraft and the ground. The command center contains two multiformat CRT display devices (which provide backup redundancy), a video monitor, a microfilm viewer, control keyboards, and selected dedicated displays and controls. To minimize the need for a full-time station operator, the control center will contain a caution and warning subsystem and a computer driven voice annunciator unit, which will provide alerting and operational information throughout the Space Station via the audio system. This permits complete monitoring safety and alerting capability without the operator being required at the station. The primary command and control center is located in the crew and operations module No. 2.

The experiment and secondary command and control center is a centralized operation center for monitoring and management of the experiment programs of both attached modules and free-flying modules. This station is also capable of providing emergency and backup vehicle and subsystem control capability in case the primary command and control center becomes unuseable because of a major contingency condition. The experiment command and control center is located in an independent EC/LS area of the GPL module No. 3. Display and control hardware required at the experiment and secondary command and control center is basically the same as that required at the primary command and control center, with additional dedicated experiment displays and controls to permit monitoring and control capability over the experiment programs.

The configuration of the experiment and secondary command and control center allows for fully independent two-man operation to accommodate simultaneous experiment or subsystem and experiment activity.

The configuration of the experiment and secondary command and control center is shown in Figure 4-1. Requirements and analysis relative to the selection of the proposed configuration are summarized in the following sections.

4.1.1 Display Requirements

An analysis of display requirements has been performed to identify the quantity and types of display formats needed by the crew. The study has identified major operational functions and the associated display requirements responsive to the ground rules and mission objectives, while eliminating "nice-to-have" and other nonessential features. Both routine or scheduled activities and contingency operations have been considered. A summary of the operational functions and display formats is given in Table 4-1. Descriptions of the functions and formats are contained in the following sections.

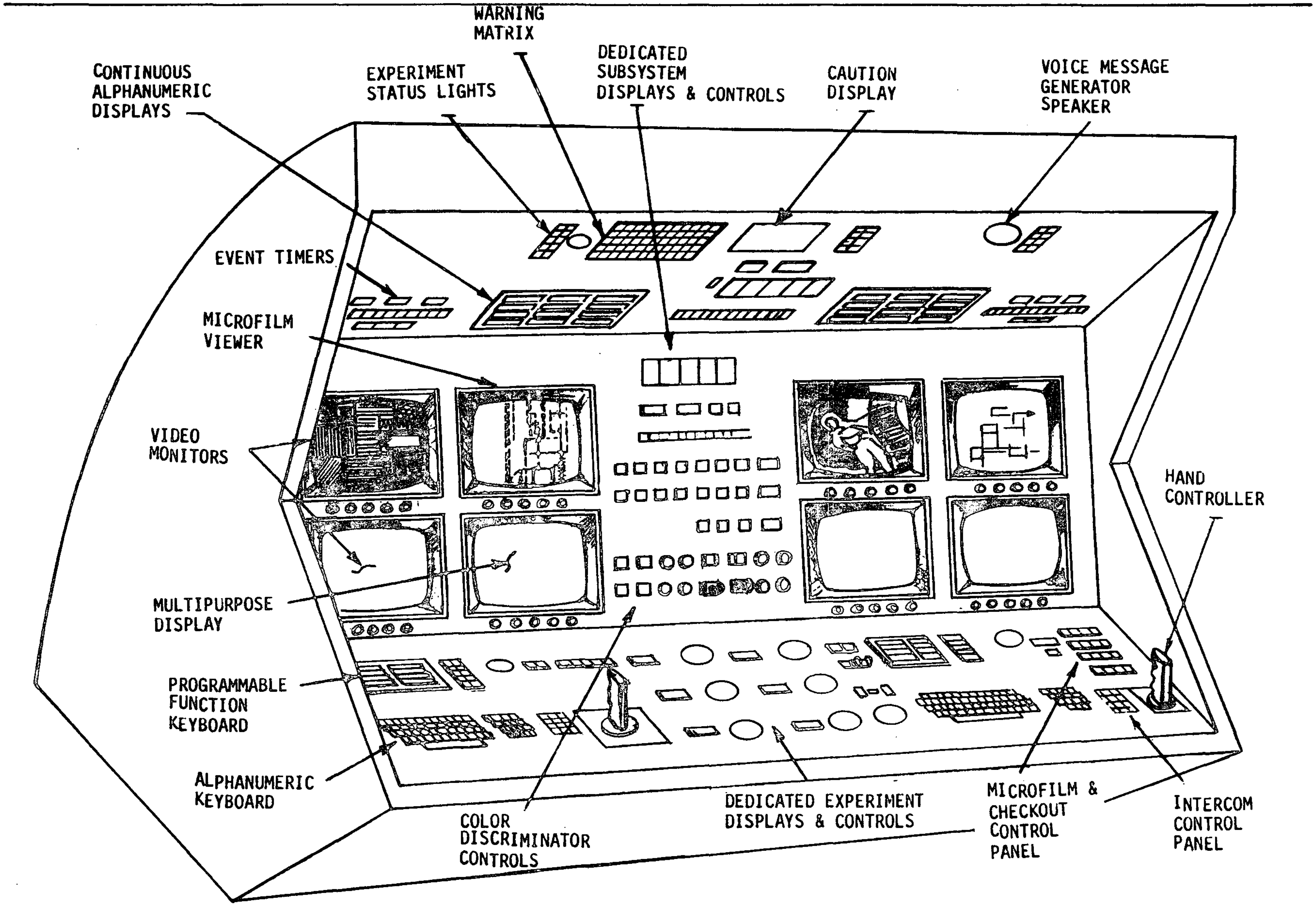

WARNING MATRIX
CONTINUOUS ALPHANUMERIC DISPLAYS
EXPERIMENT STATUS LIGHTS
DEDICATED SUBSYSTEM DISPLAYS & CONTROLS
CAUTION DISPLAY
VOICE MESSAGE GENERATOR SPEAKER
EVENT TIMERS
MICROFILM VIEWER
VIDEO MONITORS
HAND CONTROLLER
MULTIPURPOSE DISPLAY
PROGRAMMABLE FUNCTION KEYBOARD
ALPHANUMERIC KEYBOARD
COLOR DISCRIMINATOR CONTROLS
DEDICATED EXPERIMENT DISPLAYS & CONTROLS
MICROFILM & CHECKOUT CONTROL PANEL
INTERCOM CONTROL PANEL

Figure 4-1. Experiment/Secondary Command and Control Center

Table 4-1

DISPLAY FORMAT REQUIREMENTS SURVEY

	Typical Formats									
			Graphic							
Function	Text	Tabular	Bar Graph	Chart	Graph	Diagram	Line Drawing	Pictorial Overlay	Video	Micro-film
Mission Control		125	50	25	38	23	20	7		100
Flight Control		40	3		19		20	10		
Guidance and Navigation		13		4	2		8			
Maneuver Management	10	50	5	11	6	3	12	7		
Rendezvous and Docking	12	45			2	8			2	
Subsystem Operational Status		168		70	52	50			4	200
Experiments Operational Status			TBD							
Crew Status and Proficiency	180	60		12	12					420
Logistics and Resupply		75		10	10	20				
Total	202	576	58	132	141	86	60	24	6	720

4.1.1.1 Operational Functions

Nine functional catagories have been established as a result of grouping of related functions. In some cases these functions are related to specific mission phases, and in others several functions are inter-related. The categories are defined below:

Mission Control

The Modular Space Station is required to operate through onboard autonomy with little or no dependency on ground facilities. To meet this objective, the crew must have information on the location of the Space Station, the mission plan, and the mission timeline; and the capability to alter or modify the original mission as required by changing circumstances. This requires being able to call up from stored data bases, via the onboard computer system, information for planning, monitoring, and executing the desired functions. During a manned space mission, the onboard data management subsystem continually monitors mission-critical parameters and displays for the crew that data which they request in determining mission performance and progress. If a change is desired, the crew may retrieve from the data bank the current performance capability of the vehicle (i.e., what events are planned, which ones must be completed, and what unused capability remains). Assisted by this data, the crew can formulate plans and indicate events to be performed. The mechanized scheduling algorithms can program the event, determine the impact on the mission timeline, and display this information for analysis. If desired, the crew may then initiate the rescheduling of the timeline. The computer will than automatically implement the change, re-allocating required resources and subsystem capability.

Flight Control

Flight control operational functions are similar to mission control functions, but attention is focused upon the immediate situation of the vehicle. In contrast with conventional aircraft displays, the Space Station will have most of the conventional gauges, meters, indicators and other devices replaced with general-purpose CRT display and keyboard input devices. Attitude indicator displays, for example, can be "called up" and created electronically on the display. Through these displays the crew maintains awareness of the vehicle's attitude and orientation. Changes in attitude can be initiated by redirecting the flight control functions being performed by the computer. With continual monitoring and evaluation of progress, the crew can assure proper execution of the corrective maneuver. All flight control functions are normally performed via the computer, with only periodic checking and directing required by the crew.

Guidance and Navigation

The Space Station guidance and navigation (G&N) system is an automatic and autonomous system utilizing automatic data acquisition and the onboard computer for reduction and updating of G&N data parameters. The crew will be capable of manually updating or overriding computer-generated G&N parameters, and of inquiring about current vehicle position and trajectory. The crew can manually take optical and inertial data readings to update the computer for the guidance and control system.

Inquiry via the computer can provide trajectory displays, including deviations from planned course and correction requirements. Computer-driven projection map displays may be required for this function.

Maneuver Management

Because of the large degree of automation in the attitude control systems and to the sometimes precise pointing requirements on the Space Station, very little maneuvering is done manually. Display requirements are mainly limited to verification of the flight plan and trajectory analysis. In the event that the mission is not proceeding as planned or if a modification is desired, a maneuver may be required. Before performing a maneuver, the crew will perform a check on the vehicle operating capabilities and resources, utilizing the computer. This involves verifying the data base existing in the computer, initializing subsystems, determining resource availability, and establishing event sequences. After the subsystems and vehicle initial conditions are established, the maneuver will be performed, primarily by the onboard computer with manual backup and monitoring by the flight commander. During and after execution of the desired maneuver, the computer system and displays will be utilized to evaluate the maneuver success.

Rendezvous and Docking

Rendezvous and docking is a specific mission phase, which combines many functions of flight control, maneuver management, and guidance and navigation. Rendezvous and docking of a Space Shuttle is under the direction of the Space Station, although final docking is achieved by the shuttle with only monitor functions planned for the station. During this phase, the role of the station is similar to that of a ground controller at an airport. On the other hand, docking of all freeflying modules (RAM's, etc.) is by active control from the station.

Docking functions are conceived to be controlled by the onboard computer with manual backup control capability in the event of a contingency. Since the Space Station has multiple docking ports, the flight controller at the command and control station has visual contact with the incoming vehicle via television. A crew member assigned to the specific docking port will have direct vision with the incoming vehicle and will be assisted in the docking by portable display units. This crew member may take over the docking phase if it is required.

Subsystem Operational Status

Subsystem operational status and configuration management displays constitute the majority of individual frames of data to be presented. Because of the high degree of automatic control and switching in the various subsystems, the main requirement for the displays is to provide data that permit subsystem configuration selection, subsystem monitoring, and fault detection and isolation.

Experiment Status Monitoring

The primary objective of the Space Station is to perform experiments in orbit. The displays at the command and control consoles will be utilized to support these experiments and to monitor their progress. Through the data management subsystem, the computer will provide control and monitoring capability over experiment progress. The multipurpose CRT displays at the command consoles are considered adequate for general-purpose experiment usage and monitoring, but specialized and unique display equipment will also be required for many experiments.

Crew Status and Proficiency

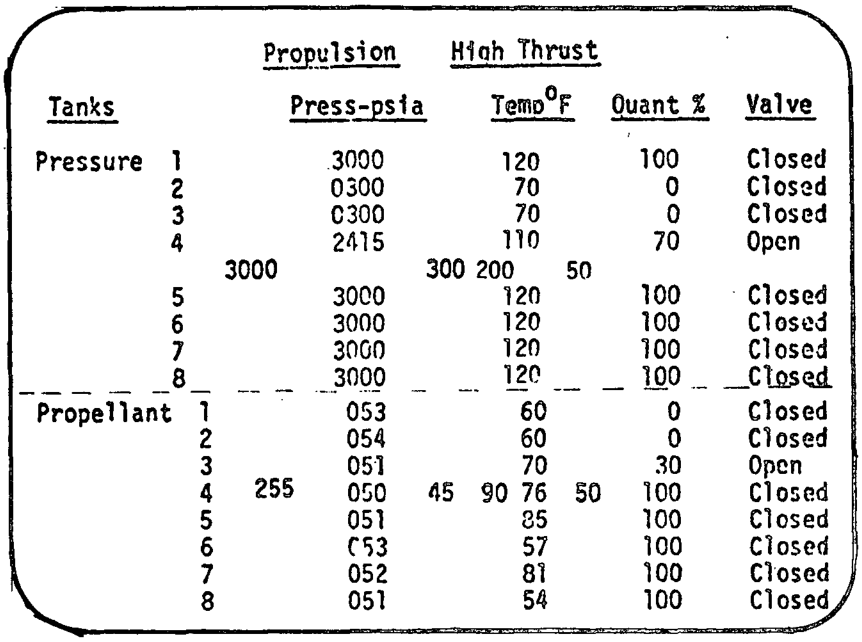
This category is a collection of mission-independent crew functions that are of a noncritical nature (i.e., delay in performing will not have an adverse effect on mission outcome). Typical functions included in this category are related to crew health status and training. Training functions are required to stimulate crew members to retain proficiency in their tasks and to develop new skills. It is anticipated that a high degree of interaction with the computer will be required since the function is heavily dependent upon simulation. Health-related functions provide updated status on the individuals physical and psychological condition.

Logistics and Resupply

Management of onboard consumables and spares and determination of resupply requirements will require the periodic assessment of the stations consumables and spares. This will be accomplished automatically to a large extent, but overall supervision and final decision making will be a crew function. This will require displays of current status and projections of future usage, utilizing tabular and graphic-type displays.

4.1.1.2 Display Formats

Figure 4-2 shows examples of typical display formats provided by CRT-type display systems. These formats consist (individually and in combinations) of alphanumeric, characters, vectors, arcs, circles, special symbols, and lines. The display may consist of static conditions; i.e., fault isolation



Propulsion High Thrust

Tanks			Press-psia		Temp°F		Quant %	Valve
Pressure	1		3000		120		100	Closed
	2		0300		70		0	Closed
	3		C300		70		0	Closed
	4		2415		110		70	Open
		3000		300	200	50		
	5		3000		120		100	Closed
	6		3000		120		100	Closed
	7		3000		120		100	Closed
	8		3000		120		100	Closed
Propellant	1		053		60		0	Closed
	2		054		60		0	Closed
	3		051		70		30	Open
	4	255	050	45 90	76	50	100	Closed
	5		051		85		100	Closed
	6		C53		57		100	Closed
	7		052		81		100	Closed
	8		051		54		100	Closed

TABULAR DISPLAY FORMAT

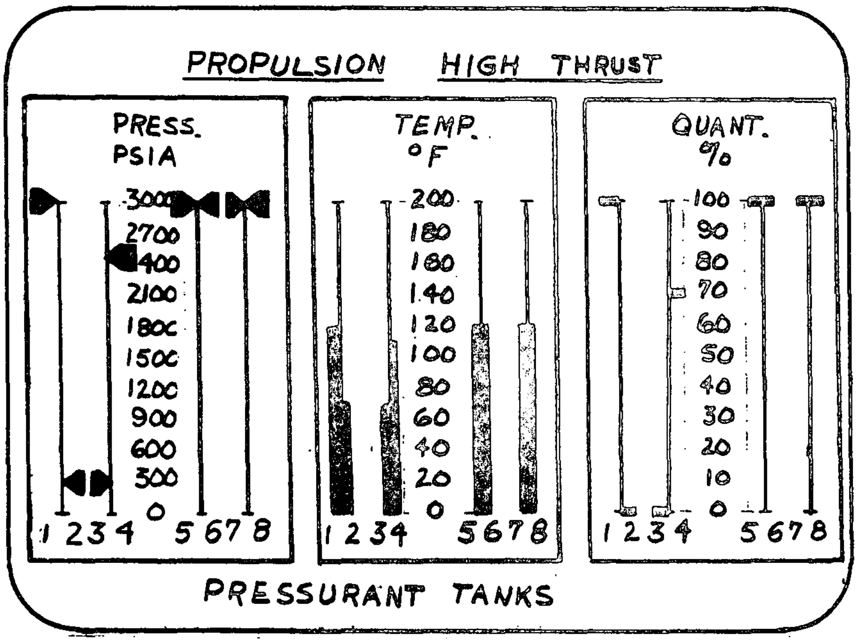
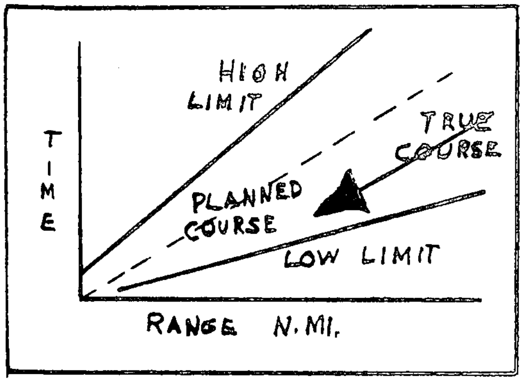


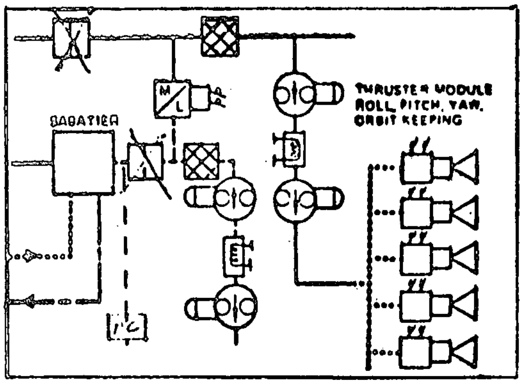
CHART DISPLAY FORMAT

(showing alternate configuration)

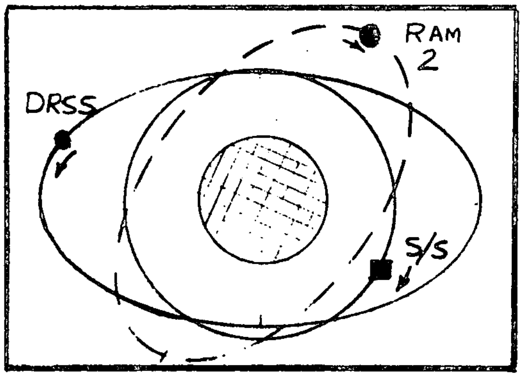
Figure 4-2. Typical Display Formats



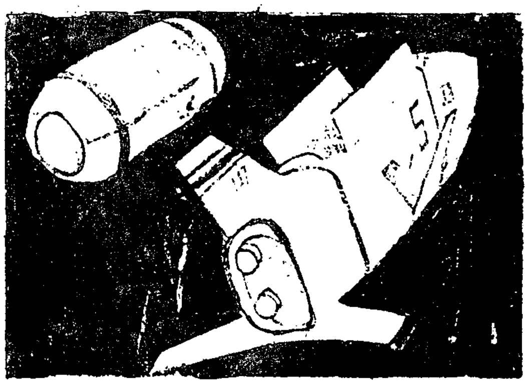
Graph Presentation



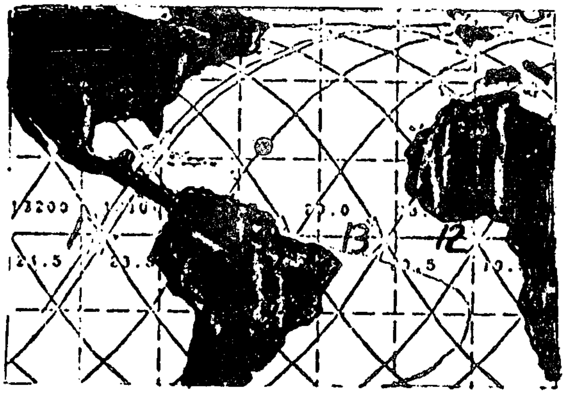
Block Diagram



Complex Graphics



Video



Dynamic Overlay Over
Microfilm Projection

Figure 4-2. Typical Display Formats (Continued)

block diagrams, as well as dynamic conditions, such as docking. Techniques for creating these displays may consist of all electronic, optical and electronic combinations, or mechanical and electronic combinations.

4.1.2 Display Implementation

This section describes the proposed display and control devices for the control centers in the Modular Space Station.

4.1.2.1 Multiformat Display

The multiformat display is the key element of the command and control center, providing as the name implies a capability for accommodating a variety of display presentations including alphanumeric, graphic, video, and combinations of these. In developing the configuration for this display, the following studies were conducted:

A. Survey of existing avionic display systems.

B. Survey of commercial multiformat displays.

C. Investigation of hardware implementation techniques.

Avionic Systems Survey

A survey of existing and proposed airborne display systems was conducted to determine system approaches and hardware types. General items of importance concerning systems utilizing a computer-display interface are noted in the following paragraphs. In many cases, however, information is incomplete because of security classification.

P3C A (New Program)—This USN program is aimed at developing an airborne display system combining sensor data, TV images, and computer information into a single display for automated antisubmarine warfare. Communication and control consoles used in this program are able to present multiple information simultaneously from a variety of inputs. Live TV and sensor information, showing ships, planes, and submarines in the area, can be combined with alphanumerics and symbols generated by the airborne computer. All of this information can be superimposed over maps to present to man. Console functions are under control of computer software and by the operator. A track ball and the controls permit him to superimpose circles, vectors, and plotting points over the displayed action. A keyboard keeps him in continuous communication with the computer. This system is similar to the commercial Datagraphix system developed by Stromberg Division of General Dynamics.

The alphanumerics and symbols displayed on a "charactron" CRT are produced by directing the electron beam through individual characters etched in a micro-matrix located within the tube. This technique "stencils" each character clearly on the tube face and permits intricate symbols to be sharply displayed.

Pictorial information, such as curves, vectors, and video displays, are produced by spot-writing methods. The micro-matrix generated alphanumerics and symbols are time-shared with the pictorial images; thus, a single display becomes multipurpose.

F-111D—This Air Force program has a limited interface between the computer and displays. Two general-purpose IBM 4 π computers for general navigations computation (GNC) and weapons delivery computer (WDC) interface with a function keyboard similar to the Apollo type. The keyboard contains function, mode select, and numeric keys with an electroluminescent segmented display register. Interface with the computer is limited to entering new data and display of subprogram processing functions in alphanumeric form.

C-5A—The C5A utilizes two computer-display systems, one for guidance and navigation and the other for onboard checkout through the malfunction detection, analysis, and repair (MDAR) system.

The conventional guidance and navigation system utilizes a digital segmented light register and numeric keyboard with function keys as the computer interface media. The man-computer communications are limited to data tolerance editing of parameters which the computer verifies and compares against aircraft performance. Should a command be given that the aircraft is incapable of performing, the computer indicates the error. The operator, after determining the command problem, clears the bad command and enters the correct one.

The checkout system monitors approximately 1900 line-replaceable units with the operator comparing "live" wave-shapes against stored microfilm wave-shapes. The display consists of a basic oscilloscope display system.

AWACS—The AWACS Program utilizes onboard analog and digital computers combined with multiple integrated display capability for display of radar and situation information. The display consoles are of identical design and contain graphic CRT's, alphanumeric keyboard, light pen, expansion control, and category switches. The display is of radar patterns with capability of adding meaningful symbols for target recognition. Computer data are also displayed in tabular and graphic form for tracking a moving target to point of intercept. The consoles are identical in configuration to meet system availability requirements and for redundancy and cost-effective reasons. In the event that one console fails, processing tasks associated with that console can be redirected to another console.

F-15—Information on the F-15 is classified and not firm, however, a CRT and alphanumeric keyboard for data display and entry is proposed. In addition, a CRT display for attitude director indication combined with TV for a heads-up display during landing is being developed.

Space Shuttle—The present Space Shuttle display and control approach utilizes multifunction CRT displays for the following functions:

Primary flight control display (PFCD)	(2)
Systems monitor display (SMD)	(1)

The display and control panel for the Space Shuttle booster is shown in Figure 4-3.

Display requirements are presently not fully defined, but they will probably include the following types of CRT display capability:

A. Alphanumeric characters
B. Vectors and circles
C. Computer Generated Tabular Data
D. Attitude display
E. TV display
F. Microfilm and mapping display

Many of the display requirements will be similar to the Space Station requirements, and common implementation and hardware may be possible since both concepts utilize multipurpose display devices. The present approach is for all display CRT's to be identical for redundancy reasons. Character and symbol generation is provided by three (individual design) display generators, with additional units for backup redundancy. Consideration is being given to a single-design, multipurpose display generator capable of driving all CRT's simultaneously. While the multipurpose symbol generator is expected to be more complex, benefits resulting from minimizing the number required and using a similar design will be lower cost, increased reliability and power effectiveness. Mapping display requirements are provided via a rear ported "window" in the CRT to permit optical projection from a microviewer projector. Electronically generated maps are also being considered as an alternative.

The baseline display system consists of the following basic hardware:

11	Digital (bus) interface units
4	Display processors
2	PFCD symbol generators
2	PFCD - SMD symbol generators
3	Display indicators (CRT)
2	Attitude display indicators (ADI) (It will probably be of a heads-up configuration.)

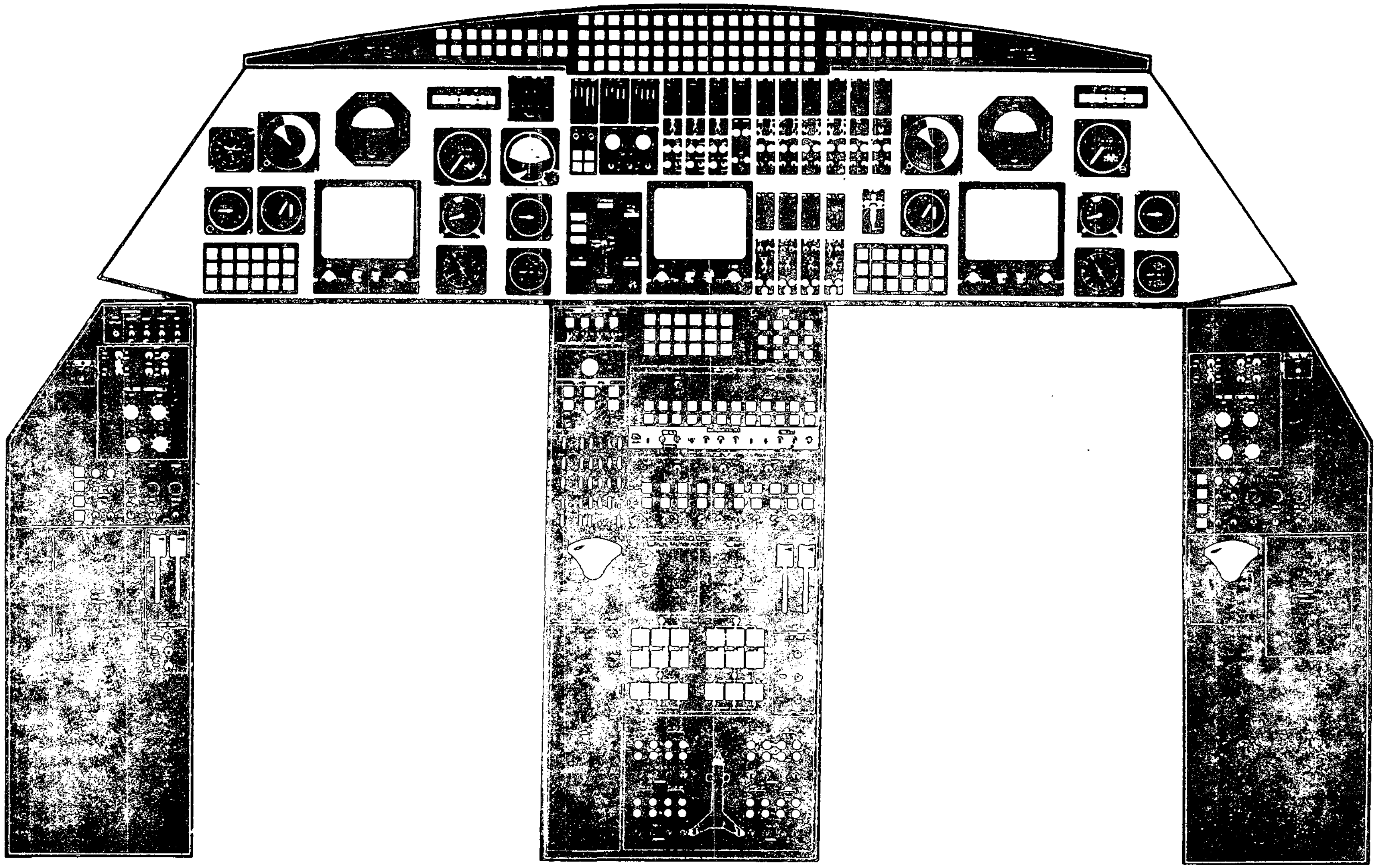


Figure 4-3. Space Shuttle Booster Instrument Panels

2 Computer keyboards - pilots and copilots
- Visual display system control panel
2 Microviewers
- Other miscellaneous display and control components

Proposed improvements include three display processors for triple redundancy and three identical multipurpose display generators capable of generating all display modes. While the multipurpose symbol generator is expected to be more complicated, the benefits resulting from minimizing both the numbers and types, together with the efficiencies that result when symbol generation functions are shared, will also result in improved effectiveness. Each symbol generator is capable of generating all displays simultaneously, and only one symbol generator is required to be on at any time. This creates a complicated symbol generator, but power consumption and reliability are expected to offset this disadvantage.

Others—A substantial amount of development in advanced state-of-the-art avionics displays is being done, aimed primarily at commercial and military aircraft applications. Typical is the system being developed by Norden Division of United Aircraft Corporation, which uses a CRT for attitude-director, horizontal-situation, and flight-deck displays. Both monochrome and color versions exist. The color system use a beam penetration tube.

Commercial Systems Survey

The survey of existing avionic systems employing CRT display technology reflects the usage of these devices generally restricted to presentation of some type of range versus azimuth data or other radar related displays. Airborne command and control situation displays or interactive computer-driven displays are virtually unknown and are a state-of-the-art area. Display consoles of this type are more prevalent in ground installations of commercial data processing complexes. A review of several commercial applications was conducted to identify salient features and design approaches. Major considerations of typical systems are shown in the following sections.

Sigma—The graphic display terminal (part of Sigma 7 system) provides a means of dynamic visual interaction of man with computer via combined graphic and alphanumeric CRT display. The terminal contains the display CRT, 64 ASCII character alphanumeric keyboard, function keys, and a light pen cursor to identify coordinates of a specific displayed item of interest.

The 21-inch (outside diameter) CRT provides a usable display area of 10 by 10 inches and employs electromagnetic deflection with electrostatic focusing. This viewing area accommodates 1024 positioning points in both X and Y axis.

The display generators consist of two vector generators, one character generator (three sizes), one dot generator, and one raster generator.

The two vector generators permit long and short vectors, with maximum lengths of 10 inches (full screen) and 1/4 inch, respectively. Short vectors are drawn in 6 μsec, and long vectors are drawn in 41 μsec.

The character generator utilizes the stroke technique and permits 64 characters to be generated in three height sizes: approximately 5/32, 5/16, and 5/8 inch high. Characters can be positioned randomly or in adjacent positions to form horizontally placed character combinations. Character drawing time is directly proportional to the size of characters, thereby providing an undistorted, constant-intensity alphanumeric display.

The dot generator allows dots to be plotted on the screen at speeds of 9 to 15 μsec per point, depending on the separation of dots.

The alphanumeric keyboard allows input of the entire set of 64 displayable characters. It also contains five special cursor-motion keys. The function keyboard consists of 16 programmable function keys and four interrupt-generating keys.

A raster generator allows the unit to detect the position of the light gun when the gun is pointed at unlighted areas of the display screen, eliminating the need for core-consuming tracking algorithms. The light gun consists of a photomultiplier and associated control electronics. A convenient finger switch permits the light gun to respond to light radiation emitted from the face of the CRT.

Information Displays, Inc., IDIOM—The IDOM is a fully buffered graphic CRT console utilizing simplified man-machine language for character writing, line drawing, and circle generation. Capabilities include four character sizes, four intensity levels, graphic element flashing, character rotation, and programable line structure. Operator-controlled input devices include a light pen, alphanumeric keyboard, and function switches. The output is on a 21-inch rectangular CRT. The buffer contains a programable, expandable 4096 x 16-bit random access memory. Electromagnetic deflection is used for large-scale positioning and vector writing, and a supplementary electrostatic deflection is used for character writing on the CRT.

The high-speed character generator is one of the function generators contained in the system. Characters are formed from continuous straight and curved lines. Sixty two ASCII characters are standard and are expandable to 96 symbols. Four character sizes spaced 128, 73, 48 and 36 per line are standard. Operator adjustment provides for selection character size from 3/32 to 1/2 inch.

A vector generator for continuous line generation provides four line structures (solid, dot, dashed, and dot-dashed) under program control. Intensity compensation is used for vector length to provide substantially equal brightness for vectors of any length. Continuous line generation is also utilized for circle generation. Intensity compensation is again provided for equal brightness for circles of any diameter.

The position generators, which determine the amount of beam deflection, utilize D to A converters whose resolution is 10 bits in the X access (1024 positions) and 10 bits in the Y access (1024 positions).

Sanders Associates—The model 960/10 graphic display system provides up to 128 different characters, including the standard ASCII 96 character set, as well as a variety of such vectors as solid lines, dashes, dots, or a combination of dots and dashes, all with eight programmable brightness levels. Characters may be written in four sizes, ranging from 0.125 to 0.5 inch high, with a nominal aspect ratio of 4:3. Greek, mathematical, electronic, and target symbols are available, and special symbols can be generated. The vector generator uses a constant-velocity generation technique; the character generator writes each symbol with up to 22 continuous strokes.

Commercial Systems (General)—The following items are a summation of several available commercial display systems used in general data processing applications.

Screen sizes range from about 7 by 9 inches for the majority of displays to 9 by 12 inches, with one of 18 by 14 inches being the largest investigated. Character sizes range from 1/8 to 1/4 inch and are produced by either dot matrix, stroke, or monoscope methods. Stroke-generated characters are considered less ambiguous than dot-generated characters and are easier to read. Monoscope characters are the most "natural" looking. The number of characters per line is generally between 32 and 80; the number of lines per display range from 4 to 60. Maximum number of characters displayed by any of the systems investigated is 4600. CRT units used for numeric and alphanumeric displays are of four basic types: conventional TV-type raster scan and high linearity stroke types, which require external character generation and memory; the Charactron shaped-beam CRT and monoscope, which produce the characters by an internal aperture mask and also require external memory; and the storage type which provides inherent memory within the CRT electronics and phosphors. The use of the storage tube approach eliminates the need for auxiliary buffer memories and are flicker free, but the writing speeds are very slow (2 seconds/display). Most non-storage systems use a writing and refresh rate of 40 to 60 frames/second.

In the area of keyboards, all investigated manufacturers called their units "solid-state" although they are not necessarily solid state in the usual semiconductor sense. Almost all keyboards on the market do use semiconductor logic ranging from diode matrices to LSI/MOS circuits. Only one keying mechanism uses a semiconductor device (a Hall-effect element, which changes resistance in the presence of a magnetic field) as the switching means. Other "solid-state" keyboards make use of keying mechanisms based on magnetic cores, capacitive coupling of high-frequency ac, and other proximity transducers.

Commercial Systems Summary—The majority of off-the-shelf commercial systems utilize either the TV raster scan or the stroke writing method in display generation. The raster scan method produces acceptable alphanumeric characters and is compatible with video, but it has very limited graphic capability. The stroke writing method is capable of good alphanumeric characters and can also produce high-quality complex graphics, including special effects such as pseudo three-dimensional images, but it is not compatible with video. Neither approach (nor any other single commonly used technique) is, therefore, capable of satisfying all display requirements of the Modular Space Station, which requires a combination of alphanumerics, graphics, and video.

Some "custom" systems have been designed with multimode capability. One familiar to many readers is the Saturn V Operational Display System built by Sanders Associates, which is capable of combined alphanumeric, graphic, and video display. This is a highly complex and physically large system, and is obsolescent in implementation, but it is illustrative of the required functional capabilities.

More contemporary efforts along these lines include several "digital TV" systems being produced by IBM, Data Disc, and Spatial Data, among others. These are TV raster scan systems that feature improved graphics resolution and color display capability, but they are still not comparable to the better stroke-type displays in the graphics capability.

CRT Systems and Hardware Implementation Techniques

The selection of a CRT and associated hardware depend upon many variables, some of which are:

CRT writing techniques
CRT screen size
Character size
Flicker and brightness
Black and white or color
Buffer memory

Detailed studies were conducted into each of these areas. A summary of the salient features and the recommended display for MSS is given in the following sections.

CRT Writing Techniques

Deflection techniques to position the beam and controlling its movement during "writing" of the desired information include raster scan, stroke write, and combination techniques.

Table 4-2 summarizes the features of CRT writing techniques, and Table 4-3 gives a comparison of these techniques and recommended approach.

Space Station Display Size - To determine the CRT size for the multipurpose display for Space Station, two approaches were taken:

A. An investigation of display format requirements was conducted to determine typical CRT presentations. Results of this survey show a maximum of approximately 1,200 alphanumeric characters per presentation.

B. Investigation of commercial and military systems and equipment to determine the capacity of these systems. In most cases, the systems are capable of displaying far more data than is felt to be required for Space Station displays. This is because most of them are utilized in ground-based electronic data processing applications, and speed in identification of a specific data entry is not required.

Table 4-2

CRT WRITING TECHNIQUES

Stroke Write Technique - Utilizes the CRT electron beam to sweep through the shape of each displayed symbol by series of strokes.

A. Each stroke sequentially generated

B. Each character sequential generated

C. Character quality improves with additional strokes

D. High-quality vectors, lines, and curvilinear shapes

Raster Scan Technique - Utilizes intensity modulation of the CRT electron beam, as it is systematically sweep across the face of the CRT to create the desired presentation.

A. TV presentations

B. Alphanumerics and symbols created by individual raster lines have dot-matrix appearance

C. Relative low-quality vectors, lines, and curvilinear shapes

D. Adaptable to background generation

Beam Extrusion Technique - Forms a complete character by directing the electron beam through the desired character stencil (target), where the beam takes the shape of the character. The shaped beam is then deflected to the desired location on the CRT face.

A. High-quality characters

B. Fast writing speed

C. Inflexible characters and sizes

D. No graphics capability

A maximum of 1,250 characters is therefore considered adequate for Space Station display situations, especially since there are multiple displays that can be used if required. This can be provided by a display format of 25 lines at 50 characters per line with alphanumeric characters of two sizes:

0.18 by 0.12 inch with 0.18-inch spacing between characters

0.24 by 0.16 inch with 0.18-inch spacing between characters

Character sizes are based upon a CRT viewing distance of 18 inches. This requires a useful display area of 8 by 10 inches, and a 14-inch-diagonal

Table 4-3

CRT WRITING TECHNIQUE COMPARISON

Function	Raster (TV)	Stroke	Shaped Beam
Character quality	Moderate	High	Best
Graphic capability	Limited	Yes	No
Graphic quality	Poor	High	--
Video compatibility	Yes	No	No
Character generator complexity	High	Moderate	Inherent in CRT
Display indicator complexity	Low	Moderate	Moderate
Power	Equivalent	Equivalent	Lower
Weight	Equivalent	Equivalent	Slightly lower

Conclusions

A. No single technique for a multipurpose display having acceptable alphanumeric character quality, graphics quality, and video capability is apparent.

B. Best approach is for dual-mode display (raster scan for video with alphanumerics and graphics generated by stroke technique during the vertical retrace period).

C. Alternative approach is to separate MSS display requirements into two display devices: (1) Stroke-write unit for alphanumeric characters, graphics, and simple background overlays generated from computer storage, and (2) Raster scan unit for TV surveillance and video, and complex background overlays with limited alphanumerics and vectors. Overlays are generated in video format and mixed with dynamic data.

CRT adequately provides this display screen area. Graphical presentations of 7 by 9 inches with adequate resolution at the edges are also provided with this size.

Display Brightness and Flicker — Nonstorage CRT's require updating of the data on the screen at sufficient rates to eliminate the phenomenon of display flicker. Flickering is a composite effect of the following functions:

Ambient lighting intensity
Display intensity
Data refresh rate
Observer

Flicker is directly related to ambient lighting; the brighter the ambient incident lighting, the more noticeable flicker becomes with fixed screen brightness and refresh rate. An optimum relation exists between ambient lighting, screen brightness, and data-refresh rate. Optimum conditions are for the ambient lighting to be low (about 3 to 7 ft lamberts) consistent with safety requirements, display brightness to be at 20-to-40 ft lambert level to prevent the CRT from excessively fast deterioration while providing soft, nonfatigueing viewing, and for the updating (refresh) to be at as low a level as possible to minimize system requirements related to data transfer rate, memory size, and buffering constraints. A refresh rate of 40 times/ second has been selected.

The use of CRT displays in the Space Station may present some special illumination problems as the operator may shift from the displays which have low surrounding ambient illumination to other areas having higher illumination levels. Various possible solutions include provisions for locally controlling illumination at the displays, use of limited-spectrum light with appropriate filters, and the use of polarized lighting.

Data Coding and Color

Data coding is primarily used for three reasons:

A. Aid in target discrimination or functional identification.
B. Indicate relationship between displays.
C. Show critical conditions.

In other words, coding should be used to distinguish between types of data, to show relationships between data, and to show the importance of data.

The following techniques are used in coding functions:

A. Use of upper-case alphanumerics to stress relative importance.

B. Higher-intensity levels for more important or critical items.

C. Intermixing of solid lines, dashed lines, and dotted lines for graphics..

D. Positioning of display information into areas reflecting importance.

E. Use of shaded areas, boxed-in data, or special outlining of critical data.

F. Use of blink or flash. Used primarily as an attention-getting device, especially when an immediate response is desired. Blink refers to periodically cutting off the symbol; flash to periodically increasing the brightness.

G. Grouping of data by color. Tables 4-4 and 4-5 summarize color generation techniques and comparison with black and white.

It is the conclusion of this study that, based upon present and forecast state of the art, the advantages of using color for general display applications are not sufficient to offset the disadvantages; namely, loss of brightness and resolution and increased complexity. Special requirements relative to experiment image processing have been idenfified, however, that indicate the need for color displays for these applications.

Display Memory—Display systems providing synthetic presentations, such as situation and tabular readouts differ from real-time systems, (such as radar and TV) by requiring a memory. This memory is used to store data between data changes, and allows refreshing of the display at sufficiently high rates to prevent flicker. The memory may be located in the CRT (via a storage tube), in the display console (via a storage drum, core, delay line etc.), or in the computer proper.

Table 4-4 (page 1 of 3)

CRT COLOR GENERATION TECHNIQUES SUMMARY

Color Generation Technique	Operational Principles	Color Presentation Characteristics	Tube Vendors	Remarks
Perforated Shadow Mask CRT	3-gun, 3-phospher Dot mask and dot phospher pattern Color by dot phospher selection	Full color spectrum 100 ft-lamberts 525 lines	Many	Extremely hard to ruggedize. Poor resolution (20 lines/in.) Relatively low brightness. No stroke write capability.
Trinitron Aperature grill CRT	Single-gun, 3-phospher Vertical phospher strips Color by strip phospher selection	3 colors 100 to 200 ft-lamberts 500 lines	Sony	Extremely hard to ruggedize. Better resolution and brightness than shadow mask. No stroke-write capability.
	3-gun, 3-phospher Vertical grill mask and strip phospher pattern Color by strip phospher selection	Full-color spectrum		Same
Current Modulation CRT	Single gun Homogeneous mixture of linear and nonlinear phosphers	3 colors (typical green, yellow, red) 100 ft-lamberts 500 lines	Sony Litton ITT	Extremely difficult phospher manufacturing problem. Interdependence of brightness and color

Table 4-4 (page 2 of 3)

CRT COLOR GENERATION TECHNIQUES SUMMARY

Color Generation Technique	Operational Principles	Color Presentation Characteristics	Tube Vendors	Remarks
Current Modulation CRT (Continued)	Color by beam current modulation			Laboratory development stage only. Rugged Compatible with solid-state electronics
Tonotron	Direct-view storage tube Two phosphers Selective erasure	2 colors 100 to 200 ft-lamberts 250 lines	Hughes	Complex high-voltage switching Erasure time limitation
Beam Penetration CRT (1) 1 gun, 2 phospher	1 gun 2 layers of phospher Color by high-voltage (anode) modulation	3 colors (typically green, yellow, red) 80 to 150 ft-lamberts 1, 000 lines	Sylvania GE Thomas	Practical airborne implementation
(2) 2 gun, 2 phospher	2 gun 2 layers of phospher Color by gun selection	2 colors 80 to 150 ft-lamberts 1, 000 lines		Not developed

Table 4-4 (page 3 of 3)

CRT COLOR GENERATION TECHNIQUES SUMMARY

Color Generation Technique	Operational Principles	Color Presentation Characteristics	Tube Vendors	Remarks
Beam Penetration CRT (Continued)				
(3) 1 gun, 3 phospher	1 gun 3 layers of phospher Color by high-voltage modulation	3 colors 50 to 100 ft-lamberts		Not developed
Banana Tube	Single gun and rotating drum Beam spot wobble	3 colors 20 to 40 ft-lamberts Very low resolution		Severe mechanical Problems in airborne Low brightness
Goodman Tube	Light/voltage light pipe CRT	3 colors 50 to 75 ft-lamberts 600 lines		Low brightness
Apple Tube	Beam indexing	3 colors 350 lines	RCA Philco	Poor color purity
Chromatron CRT	3 gun, 3 phospher Vertical phospher strips	Full color spectrum 300 ft-lamberts Low resolution	Sony Litton	No stroke write capability

Table 4-5

COLOR VERSUS MONOCHROME CRT TRADE STUDY SUMMARY

Comparison Parameter	Beam Penetration	Monochrome CRT
Writing technique	Raster or stroke (only compatible with monochrom TV camera)	Raster or stroke

Brightness	Green = 500 ft-lamberts
	Yellow = 150 ft-lamberts
	Red = 50 ft-lamberts
		1,000 to 2,000 ft-lamberts
		Typical—7 shades of gray

Resolution	75 lines/inch (green) 50 lines/inch (red)	100 lines/inch
Linearity and registration	Comparable	Comparable
Viewability	Fiber optics filter required: viewing angle is ±20 degrees	
Reliability	High-voltage switching Additional circuitry and complexity	
Cost	20 percent increase per indicator 0.1 percent increase in symbol generator	
Operational advantages	Determine by simulation	

If the computer memory serves as the display memory, the computer has absolute control over what data are to be displayed, where the data are stored, and what information is updated. On the other hand, the computer-to-display transmission link will be in use constantly at a relative high rate to provide the necessary data to the CRT on a flicker-free basis. For example, if the display data are contained in 1,000 words and the desired refresh rate is 40 cps, a new display word must be transmitted every 25 microseconds. With the total number of individual displays that may be operated at the same time (two at the primary station, two at the secondary station and three portable), an excessive amount of computer time and data link capacity would be required just for servicing this input/output function. Because of this it is concluded that the display memory should not utilize the central computer's memory, but have local memories that are updated only as data changes or requests for new data are made.

State-of-Art Display Review

A survey and review of technical literature was conducted into state-of-art display technology and devices to determine the feasibility of a "flat panel" display as a CRT replacement unit. Considerable research is being conducted into this area, and three of the more promising ones are listed in Table 4-6, with their basic capabilities.

Light emitting diode (LED) technology is the more advanced within the flat panel display field at this time. Equipment manufacturers are concentrating upon the area of discrete status lights, pilot lights, and numerical displays consisting of few digits. The primary factors preventing wider use are the relative high cost of LED devices and drive electronics and the mechanical problems associated with fabrication of large arrays. The seven-segment hybrid (numeric only) display device is the most common type of LED readout. Seven-segment monolithic units use considerable more material and thereby increase cost, and dot matrices (having alphanumeric capability) require additional drive electronics which also increase costs. The use of a 16-segment hybrid device for full alphanumeric capability appears to be an ideal compromise for displays of limited alphanumeric characters (less than

6

Table 4-6
DISPLAY TECHNOLOGY REVIEW

State-Of-Art Techniques

Gas discharge plasma panels - array of gas cells triggered by voltage applied to appropriate row and column in X-Y grid.

A. Alphanumerics and graphics via programmable dot matrix

B. Gray scale capability

C. Limited color capability

D. Translucent panel permits rear projection of microfilm

E. Presently 60 elements/in. (expected capability: 100 elements/in.)

Light emitting diode (LED) - array of diode modules arranged to form flat panel display.

A. Numerals only at present time

B. Red color only—green color in prototype stage

C. Extremely long life - rugged construction

D. Low-voltage LSI electronics address

E. Relative high cost per bit

Liquid crystal - clear organic liquids within a glass sandwich that becomes opaque when exposed to an electric field.

A. Selective application of voltage to conductive film generates character

B. Microwatt power dissipation

C. Low cost

D. Ease of manufacturing

E. Temperature and material limits

100 characters). Red will remain the dominant color of LED readouts. Green emitting diodes suitable for numeric displays are still in the laboratory stage.

Advantages of LED devices are: extremely long lifetime, ruggedness, low voltage requirements, and compatibility with integrated circuit electronics.

Disadvantages include: excessive heat dissipation from panels containing large numbers of lighted segments or dots, moderate power requirements, large number of electronic driver circuits requiring power, and cost of manufacturing panels of LED chips.

Large flat panel displays of the future will probably utilize gas discharge plasma panels or liquid crystal cell technology. Plasma panels, which are available today in moderately large sizes, consist of a sandwich construction of glass panels containing multiple rows of tiny gas filled cavities. Under proper voltage stimulation from an X-Y electrode grid, a gas discharge is created within the cavities to produce a dot glow. A pattern of these dots are utilized to create dot matrix characters or graphical presentations. Through use of multiplexing signals to the appropriate row and column grids in the panel, electronic driver circuits are considerably reduced. Gray scale capability is also achieved through multiplexing of the appropriate cells on a time-duration cycle, which creates a variable brightness effect. A disadvantage of the plasma panel is that it requires an excitation voltage of 250 volts, which makes the driver circuits incompatible with LSI circuitry.

The liquid crystal technology is the newest and possibly has the greatest potential of the three techniques investigated. Liquid crystal displays use a class of liquid crystals called "nematic," which are clear organic liquids that become opaque when exposed to an electric field. Displays are made by putting a thin layer of liquid crystal between two plates of glass, one of which is completely coated with a conductive film and the other coated in a pattern, such as the seven segments of a numeric readout. Electric fields can then be applied selectively to form characters. Excitation voltages are low, making LSI type electronic circuits directly compatible. Since the glass/liquid crystal/glass sandwich has very high resistance, power dissipation is extremely low—in the microwatt region; and since the display operates by reflecting and not by generating light, the brighter the ambient light is, the better the visibility of the readout.

Liquid crystal material is inexpensive and fabrication uncomplicated because the change in reflectivity is a bulk, and not a junction, phenomenon. As a result, liquid crystal displays are inherently inexpensive.

There are problems, however. One is temperature; liquid crystal material tried to date must be kept above 0° C and below 70° C. Another is the long response time of the material; it requires about 200 milliseconds to go from the opaque to the clear state, and the eye can see this slow turnoff. Moreover, operating lifetime is limited - 10,000 hours when run with an ac field.

All of these problems relate to the electrochemical properties of the liquid crystal material. Researchers feel that the solution largely amounts to coming up with the right compound, and if these limitations are overcome, this technology may be a prime contender for the Space Station display devices.

Multiformat Display Configuration Summary

The selected configuration for the multiformat display is a CRT device that is capable of presenting computer-generated alphanumeric and graphic data, as well as video imagery from vidicon cameras or other sources. The computer-generated and video data may be shown independently, adjacent (split screen), or superimposed to provide complete flexibility in display formats. The display will operate in one of three modes:

A. Stroke-write mode for displaying computer-generated data, including alphanumeric characters, vectors, lines, and graphic presentations.

B. Raster scan technique for displaying TV imagery and other analog input data.

C. Dual mode technique for mixing of alphanumeric data with a TV image. This is achieved by allowing the display to create the TV image in a normal raster scan mode with the display going into the stroke-write mode during the vertical retrace time period of the normal raster scan sweep.

The display system includes a full alphanumeric keyboard for operator data inputs, and a hand-held light pen. The light pen may be used in conjunction with the CRT display for a variety of interactive functions, including control and data or image processing.

A buffer memory is provided for the storage of computer-generated data. This permits once-only transmission of data from the computer to the display for each new (or changed) data frame, thus unburdening the computer and the data bus. Periodic refresh of the display is performed from the buffer memory.

A summary of multiformat display characteristics is given in Table 4-7.

Table 4-7
MULTIFORMAT DISPLAY CHARACTERISTICS

- 96 ASCII alphanumeric character set
  - Upper and lower case alphabet
  - 0-9 numerals
  - Special symbols
  - Graphical designations
  - Function and mode keys
- 1250 characters per frame display formats
  - 25 lines with 50 characters per line
  - 2 character sizes
    - 0.18 inch x 0.12 inch with 0.10 spacing between characters
    - 0.24 inch x 0.16 inch with 0.15 inch spacing between characters
- 800 linear inch per frame of graphics
  - lines - vectors - circles - arcs - special symbols
- Local refresh memory
  - 40-Hz refresh rate
  - Memory capacity 1.25 x 10^6 bits
- Display screen area
  - Graphic presentations 8 x 10 inches
  - Alphanumeric characters 9 x 11 inches
- Video capability
  - 525 commercial standard TV lines

4.1.2.2 Video Monitor

Video monitors are provided at the command and control centers to provide internal and external surveillance capability, monitoring of incoming vehicles during docking phases, and display of experiment related data and images for on-line real-time digital image processing. The displays are driven from Vidicons installed at various locations within the Space Station modules and at the docking ports, as well as from stored memory and experiment data sensors as directed by the operator. These monitors are of two resolution levels. The general-purpose surveillances and display monitors will be 525 line-commercial standard TV capability. The monitors used for image processing will be high-resolution CRT's capable of 1029 line pairs and will have three-color capability. These monitors will have computer interactive capability and computer-generated character capability for annotation of the video display.

4.1.2.3 Voice Message Unit

A voice message generator is provided, which permits spoken-voice messages to be generated under computer control. This capability supplements the visual displays to provide caution and warning functions, as well as to provide operational and experiment control cues to the crew.

The device contains a fixed vocabulary, which is stored in digital form in the device memory. A basic vocabulary of 71 words is presently identified, and capability for expansion to 256 words (or larger, if necessary) is provided. Individual words, or phrases consisting of up to four individual words, may be selected under computer control. The selected message is then electronically synthesized and output over the Space Station audio system. The message may be broadcast throughout the station or routed to individual locations as required. The normal form of output is a male-register voice, but a female-register voice may be used if desired.

4.1.2.4 Programmable Function Keyboard

Multifunction control input capability is provided in the form of programmable function keyboard (PFK). This device features a flat panel alphanumeric display screen upon which computer generated data may be written.

These data are portioned into a series of discrete fields, each of which may be used to identify (by name) a control function available to the operator. The operator may then select and initiate actions by depressing the screen in the area displaying the desired function name. This action will cause momentary electrical contact between a series of fine X-Y grid conductors imbedded in the screen. The contact location will be encoded and transmitted to the computer, which will then take appropriate action as defined by the computer software.

The displayed list of functions may be altered by the computer to suit the requirements of the particular mission phase or operation being performed at the time. Thus, the same device may be used to control checkout operations at one point in time and to operate an experiment at another. Operator inputs to select desired display lists may be made through a series of program switches associated with the device.

An example of PFK utilization is described in some detail in Section 4.2.4 of this report. That section also contains a description and illustration of the deliverable breadboard console PFK, which is very similar to the one described here. The primary difference in implementation is that the breadboard utilizes discrete switches in place of the proposed X-Y grid wires.

4.1.2.5 Continuous Parameters Display

Alphanumeric readouts are used to continuously display selected parameters that are considered critical to Space Station integrity, important to certain mission phases, or to provide information of general interest. These readouts are displayed on a flatpanel A/N unit and interface with the computer thru the data bus.

4.1.2.6 Caution Displays

Display of caution level functions will not be via a matrix of individual annunciators but by flat panel display, which indicates messages determined by the multiprocessor. This display interfaces with the multiprocessor via the data bus and operates in a manner similar to the alphanumeric displays described in the previous paragraph. The lower portion of the display will permit storage of past caution alerts and allow recall capability of functions that have not been corrected. In this manner, status of caution functions can be determined by activating a switch to call up the message for uncorrected caution conditions.

4.1.2.7 Status Lights

Status lights are required to show subsystems, experiments, and operations conditions. These monitors will be used to indicate active or passive conditions, depict normal or alternate modes, provide positive control feedback response, and in general indicate subsystem and experiment conditions.

4.1.2.8 Dedicated Displays and Controls

Several dedicated meters and other display devices are required for unique and emergency and contingency conditions. It is expected these will be utilized in the event of emergency response, power failure, and other contingency conditions, as well as for subsystem and experiment support.

Rotary and toggle switches will be required to supplement the displays and controls. These controls may be utilized for specific subsystem and experiment functions, as well as for emergency and contingency capability. Critical control functions and backup functions will be "hardwired" for maximum reliability, while other switches will interface with the respective controlling device through the data-bus/computer link.

4.1.2.9 Other Space Station Displays and Controls

Requirements exist for other display and control devices in the Space Station that are not considered "state-of-art" and will not be duplicated in the D&C breadboard. These include:

A. Warning matrix of dedicated illuminated annunciators

B. Microfilm viewer - The microfilm retrieval system utilizes video distribution techniques and will be similar to the video monitor equipment.

C. Hand controller - Hand controller switch inputs will be similar to the other switch inputs and the proportional control capability is handled within the RDAU's for A-D conversion.

D. Analog slewing controls - same as above.

4.2 DISPLAY AND CONTROL BREADBOARD IMPLEMENTATION

The D&C breadboard console is designed to incorporate functional equipment which will operate in conjunction with a computer and the data bus to demonstrate proposed Space Station control and display concepts. The breadboard is implemented with commercial and off-the-shelf components to meet budgetary and scheduling limitations, but the equipment is representative in function, if not in form, of flight hardware. The unit contains the following items:

A. Multiformat CRT display

B. Voice message unit

C. Alphanumeric—video display

D. Programmable function keyboard

E. Light emitting diode clock and timer displays

F. Dedicated lights and Meters

In addition to the above equipment, the breadboard contains a display interface adapter (DIA), which provides the interface between the breadboard equipment and the data bus terminal.

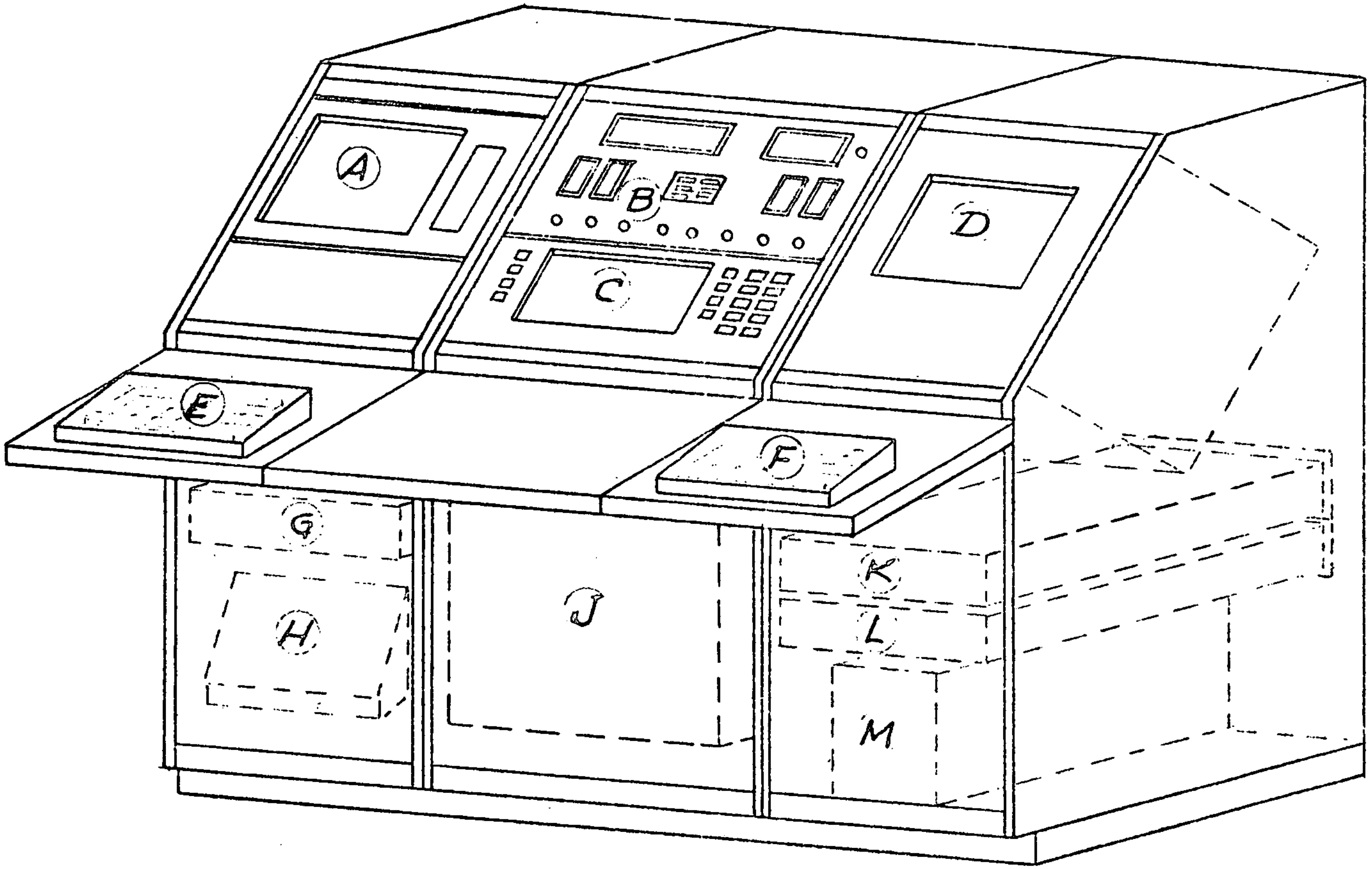
Table 4-8 lists the selected hardware by model number and manufacturer, and Figure 4-4 illustrates the console configuration.

The following sections provide a general description of the console devices and their capabilities. More complete hardware and software descriptions, operating instructions, and other detailed documentation will be supplied at the time of hardware delivery.

Table 4-8

DISPLAYS AND CONTROLS BREADBOARD—
EQUIPMENT LIST

A.	Multiformat Display Unit		
	IMLAC PDS-1		IMLAC Corporation
	High-contrast and Resolution CRT Display		
	Keyboard		
	Memory expansion to 8, 000		
	Parallel interface expansion		
	Light pen		
	Cassette bulk storage		
	Control panel		
B.	Voice message unit		ACTRON Division of MDAC
	ACI voice terminal model 3001-60 PA (prototype)		
	60 to 80 words and phrases		
	Maximum of ten new vocabulary words		
C.	Alphanumeric—video raster scan display		Computer Communications, Inc.
	Display controller—keyboard—monitor		Conrac Corporation
	Character generator		Grass Valley Group, Inc.
	External Synchronization, and video insertion		
	Protected field capability		
D.	Programmable function keyboard		MDAC
	Keyboard switches		Burroughs Corporation
	Display function array		
	Electronic and power supplies		
E.	LED display		Electronic Research Co.
	Calendar clock—10 digits		
	Timer—4 digits		
F.	Dedicated components		
	0-to 5-volt linear meters	- 4 typical	International Instruments
	Status lights	- 8 typical	
	Control switches	8 typical	
G.	Interface adapter unit		IBM
H.	Console enclosure		ENCOR Corporation



NOTE: UNITS SHOWN IN DOTTED LINES ARE ACCESSIBLE FROM REAR.

A. VIDEO-ALPHANUMERIC DISPLAY MONITOR
B. DISCRETE METERS, STATUS LIGHTS AND SWITCHES
C. PROGRAMMABLE FUNCTION KEYBOARD
D. GRAPHIC DISPLAY MONITOR
E. VIDEO-ALPHANUMERIC DISPLAY KEYBOARD
F. GRAPHIC DISPLAY KEYBOARD
G. VOICE TERMINAL EQUIPMENT

H. PDS-1 DISPLAY MAINTENANCE CONTROL PANEL AND CASSETTE RECORDER (STORAGE)
J. PDS-1 DISPLAY PROCESSOR
K. VIDEO INSERTION EQUIPMENT
L. DISPLAY INTERFACE ADAPTER UNIT

Figure 4-4. Display and Control Console Breadboard

4.2.1 Multiformat Display

The device selected for the multiformat CRT display is a PDS-1 Computer Display Unit, manufactured by IMLAC Corporation. The PDS-1 includes a CRT display screen, a keyboard, and a light pen for the operator interface, and is capable of displaying both graphic and alphanumeric data, using stroke writing techniques.

Display generation is controlled by a software program residing in a small self-contained general purpose mini-computer. The mini-computer includes an 8, 128 word memory unit, which is used to store a listing of the characters and line segments that make up a given display presentation, and which also contains a series of coded instructions that specifically define the sequential CRT beam movements necessary to trace out each displayable character. These codes control a separate special purpose Display Processor, which converts the codes into beam deflection voltages. Thus, to display a given character the minicomputer issues, an instruction to the display processor, identifying the desired character. The display processor then executes the instruction, accessing core for the specific display codes associated with that character. Operation of the display processor during execution of the instruction is completely independent of the minicomputer, except that the display processor steals memory cycles when accessing core. The minicomputer in the meantime is free for other tasks, such as processing keyboard data entries and data bus input/output operations. This method of display generation provides tremendous flexibility in that the designs of individual characters can be changed by software, and similarly, entire new character sets and arbitrary symbols can be created. A standard software package, called Symbol Form, is provided with the PDS-1 to accomplish this.

The PDS-1 interfaces with the data bus via the display interface adapter. The interface transfers 16-bit parallel words on both input and output at a nominal rate of 55,500 words per second (burst rate), which matches the Data Bus Terminal input/output transfer rate. The maximum transfer rate is approximately 100,000 words per second. It should be noted, however, that all I/O transfers are restricted by the data bus design to blocks of 32 words maximum.

Actual transfer rates and other I/O functions are under control of the mini-processor software. Specifically, these functions are included in the edit program, which is supplied with the PDS-1. This program also includes the routines associated with processing of keyboard inputs and performing text manipulations such as character insertion and deletion, and with conversion of input data to the appropriate display instructions. All alphanumeric data transfers are in standard ASCII format.

Two pieces of accessory equipment are provided with the console to allow control and programming of the minicomputer. The first is a small control panel that allows manual entry of data to the computer and also allows visual readout (via lights) of register and memory contents. This panel is intended for use primarily as a maintenance aid. The second item is a magnetic tape cassette storage unit for loading the core memory and for storage of core contents as desired. Both the control panel and the cassette unit are plug-connected and are normally disconnected and stored when not in use.

Provision is also made for connection of a standard teletype console to the minicomputer, although the teletype itself is not provided. This feature is useful, however, as the teletype provides a convenient means of programing the device. Of particular interest to the programmer are the standard assembly programs that are supplied with the machine and which are designed to operate with the teletype unit as an input/output device.

A summary of the PDS-1 characteristics is shown in Table 4-9. Characteristics of the miniprocessor are given in Table 4-10.

4.2.2 Alphanumeric/Video Display

The alphanumeric/video display system is capable of outputing alphanumeric and vidicon type displays, either separately or in combination. The system includes a display monitor, a display controller and keyboard, and video sync and insertion equipment. Key features are summarized in Table 4-11. A block diagram of the system is shown in Figure 4-5.

Table 4-9

SUMMARY OF PDS-1 DISPLAY CHARACTERISTICS

CRT Specifications	Description
Tube size	14-in. diagonal
Phosphor	P 39.
Spot diameter	12-mil nominal, high-contrast, high-resolution tube
Deflection method	Magnetic
Focus method	Electrostatic
Brightness	50 foot lamberts
Contrast ratio	20:1, high-contrast tube
Refresh rate	40 frames/sec, (60 frames and 30 frames, optional)
Display Method	
Deflection type	Random scan - stroke write
Vector generation technique	8-bit bytes comprising character subroutines are decoded by hardware as display vector strokes; dual D/A's per axis drive deflection amplifiers.
Character generation technique	Character symbols are composed of a series of software controlled vector strokes. The electron beam is turned on and off during character drawings.
Display size (w x h)	8 in. wide. 8 in. high
Characters per line	128 maximum with horizontal tube hardware. Software adjustable.
Total displayable characters	About 2,000, flicker-free, on an 8,000 system at 40 frames per sec, depending on character complexity.

Table 4-9

SUMMARY OF PDS-1 DISPLAY CHARACTERISTICS (Continued)

Display Method (Cont)	Description
Character size: assuming standard 80 char/line adjustments; different sizes can be achieved at other character densities.	Programable; with standard text and edit program, scale 1, Upper case alpha 1/8 in. high.
Number of lines of characters	40 standard; 64 maximum
Fonts: (Standard font is upper and lower case 7 by 9 vector stroke matrix)	Completely determined by software routines; therefore, it is possible to create and display any special symbols.
Maximum Resolution (in X and Y)	1024 by 1024 addressable points
Input devices	
Keyboard	Similar to standard teletypewriter but completely software oriented; the codes generated by the keys can be interpreted by software to perform any function.
Function and editing keys	Standard, keyboard, software programable.
Light pen	Generates interrupt and stores address
Graphics input capability	Standard from keyboard; light pen and host computer.

Table 4-10

PDS-1 MINIPROCESSOR HIGHLIGHTS

FEATURE	DESCRIPTION
MEMORY	
Word size (bits)	16
Memory size (words)	4-k standard; 32-k optional, by 4-k sections. Delivered system will contain 8-k
Cycle time (μsec)	2.0
Memory protect	Optional-by 2-k sections
Direct addressing (words)	2K
Indirect addressing (words) (set bit zero)	up to 32-k single level
MPU	
General purpose registers	1 + link bit for multiple precision
Index registers	8 auto-index in locations $10_8$to 17_8 in each 2-k block of memory
Immediate instructions	yes
I/O (parallel)	
I/O word size (bits)	16
Priority interrupt levels	1
I/O maximum word rate	100-k words without checking
OTHER FEATURES	
Power fail/restart	standard
Core protect	standard feature; special circuitry prevents any modification in core memory due to power interruption. Thus, if display is on the screen when power is lost, display will reappear with restoration of power.
Read only memory bootstrap (ROM)	Included

Table 4-11

ALPHANUMERIC/VIDEO DISPLAY FEATURES

ALPHANUMERIC DISPLAY

Baseline unit—Computer Communications, Inc., Model CC-30

MAJOR FEATURES

Character—graph generator
1024 words—9-bit random access core memory
Raster scan commercial TV compatible monitor
I/O for computer communications
Alphanumeric and control keyboard
Light pen
50,000 bits per second serial-transmission rate

OTHER FEATURES

14-in. diagonal CRT
12- by 10-in. Screen Area
20 lines of 40 characters per line
5 x 7 dot matrix characters
Refresh rate 60 Hz
External sync capability
Video insertion/mixing

4.2.2.1 Display Monitor

The display monitor is a standard Conrac 14-inch monochrome video monitor capable of accepting a 525-line composite video signal or separate video and sync. The monitor is a transistorized unit having high reliability and low power dissipation. The video response is flat to 10 MHz and provides a CRT center resolution of up to 800 lines. Corner resolution is 600 lines. Front-panel controls, such as ON and OFF, horizontal and vertical hold, brightness, and contrast, are enclosed under an access door.

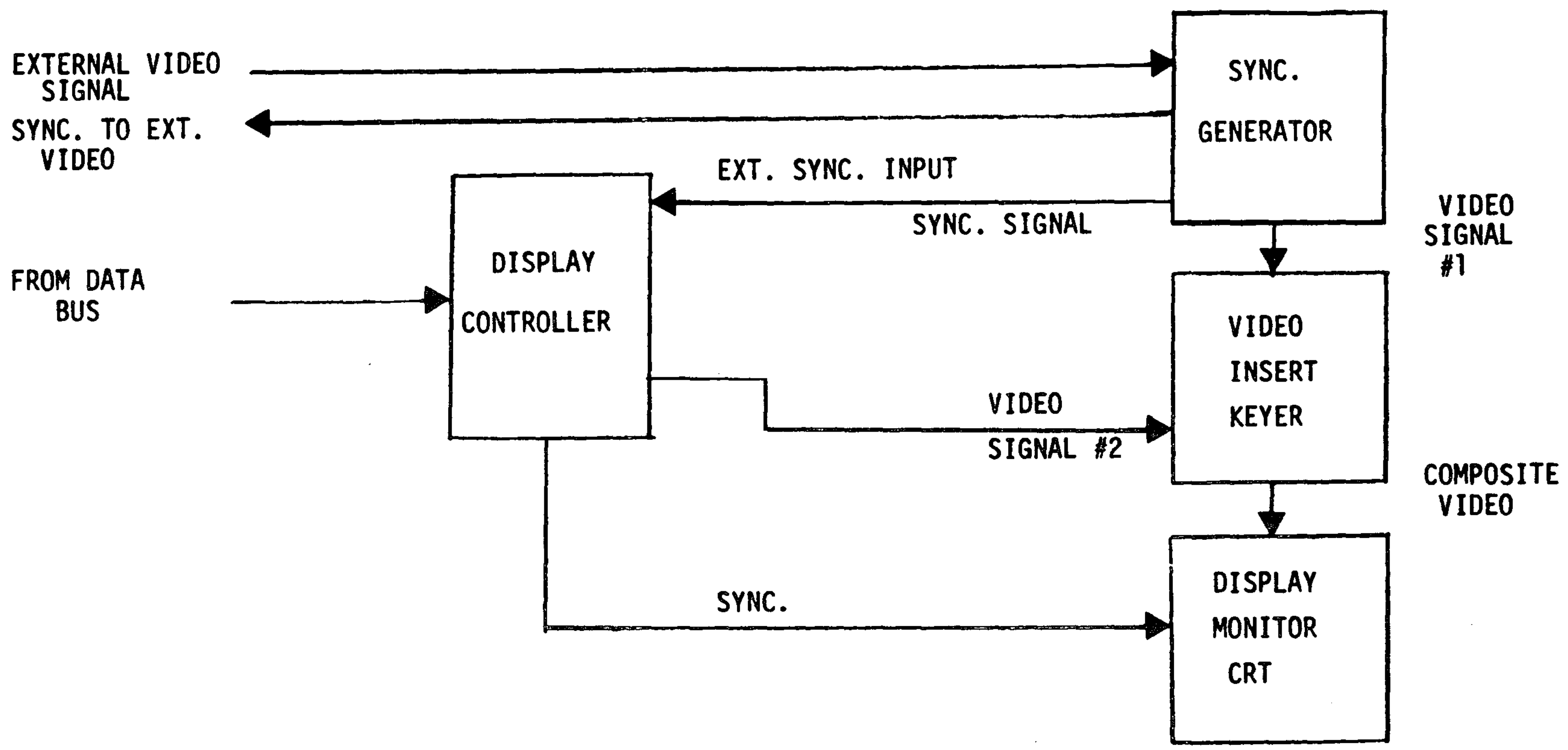


Figure 4-5. Alphanumeric/Video System Block Diagram

4.2.2.2 Display Processor and Keyboard

The alphanumeric information is generated by a Computer Communications model CC-30 terminal, consisting of a CC-301 display controller and a CC-303 Alphanumeric Keyboard. The controller is capable of receiving ASCII coded data inputs from the data bus (via the DIA) or from the keyboard, and of transmitting keyboard-entered data to the data bus (again via the DIA). The data bus interface transfers are in the form of 8-bit parallel words at a maximum rate of 500,000 words per second. Nominal transfer rate is limited by the data bus terminal to approximately 125,000 words per second. Data input to the controller is stored in an internal core memory that provides buffer storage and refresh capability. The controller outputs data to the display monitor in standard 525-line video form. External video sync is supplied by the video sync and insertion equipment. The controller has two modes of operation: alphanumeric and graphic.

In the alphanumeric mode, characters stored in the memory as ASCII codes are displayed on the TV screen in a format of 20 lines of 40 characters each. The memory locations whose contents are not displayed are accessible to the computer and can be used for storage of such information as terminal identification, accounting data, or for on-line diagnostic purposes. This memory can also be used as a buffer area for communication between the computer and any of the optional input/output devices under the control of a cursor (entry marker), which always indicates the location of the next memory access. The cursor appears on the screen as a horizontal underline and marks the position that the next character will occupy upon entry into memory (from either the computer or a local input/output device). As characters are transmitted into or out of memory, the cursor is automatically advanced to the next sequential character position, including advancement to the beginning of a new line when necessary. The cursor symbol, since it occupies the display area below the character, can be moved freely about the screen without erasing or disturbing displayed characters. The contents of memory are continuously displayed and can be transmitted at any time to a computer or optional local output device by either operator or computer command. The transmission does not destroy the contents of memory.

In the graphic mode, data are displayed by means of a 108-by 85 matrix of dots. Each dot is uniquely represented by a single bit in the first 1,020 words words of memory. One nine-bit memory word corresponds to nine successive dot positions on a horizontal scan line.

The capabilities of the display processor may be expanded through the addition of optional equipment. The available options include a light pen, card reader, line printer, teletypewriter, and telephone coupler. An option is also available to allow the generation of displayed information in up to four intermixed colors (green, red, blue, and yellow).

4.2.2.3 Video Insertion

Capability for mixing computer-generated alphanumeric data and external video is provided by a Grass Valley Group, Inc., insertion keyer and syncronous generator. This allows an external video signal, such as from a vidicon camera, video recorder, or flying spot scanner, to be mixed with the character information in such a way that the characters appear on the display monitor superimposed over the video picture. It is also possible to display either the external video or the alphanumeric information without mixing. The syncronous generator provides the video sync signal necessary to synchronize the external video source, the display controller, and the monitor.

4.2.3 Voice Message Generator

A voice message generator (voice terminal equipment) designed and built by Actron Industries (a division of MDAC) is furnished as part of the display and control breadboard for use as an integral part of the caution and warning system as well as for general mission operations and experiment interaction. This device provides a direct computer-to-man voice communications link through the audio system, using a digitally controlled voice synthesizer. Table 4-12 summarizes the device characteristics.

The voice terminal equipment (VTE) is a solid-state voice synthesizer having a digitally stored vocabulary which generates clear, distinct voice signals on command by digital codes. The digital code selects a

Table 4-12

VOICE MESSAGE GENERATOR CHARACTERISTICS

Functions

A. Provides advisory and alerting information

B. Identifies status and malfunction notification

C. Reduces visual task loading—incorporates auditory sense

D. Permits unmanned control center operation

E. Supplement/back-up for monitors and displays

F. Direct machine to man communications

Features

A. Digital-stored and addressed vocabulary

B. All solid state

C. 60 words and phases of vocabulary via ROM memory

D. Modular expansion and change of vocabulary via ROM memory

E. Rapid and inexpensive vocabulary change or augmentation

F. 8-bit addressing code

G. Greatly improved reliability

particular message or individual message or individual word from the stored vocabulary contained within the voice indicator. The vocabulary is stored in read-only memory (ROM) arrays, which permit rapid and inexpensive change or augmentation of vocabularies.

The voice terminal equipment contains control logic and mounting provisions for 16 ROM arrays, containing a total of about 48,000 bits of storage. This represents 71 spoken English words, which may be stored in any combination of individually selectable words or short messages containing two to five words. Any word or message is selected by an 8-bit address code. Thus, the computer may compose messages by a sequence of 8-bit codes, which select the desired words or may address a fixed message by a single 8-bit code. These two modes may be mixed in any way. Table 4-13 lists the selected vocabulary.

Table 4-13

VOICE TERMINAL VOCABULARY LIST

1. One
2. Two
3. Three
4. Four
5. Five
6. Six
7. Seven
8. Eight
9. Niner
10. Zero
11. AC
12. Battery
13. Bus
14. Cabin
15. CO_2
16. Communication
17. DC
18. Display
19. High
20. Low
21. Multiple
22. Oxygen
23. Pressure
24. Primary
25. Propulsion
26. Power
27. Secondary
28. System
29. Warning
30. Caution
31. Alignment
32. Auto
33. Computer
34. Crew
35. Emergency
36. Go
37. Hold
38. Leak
39. Limit
40. Mode
41. Overload
42. Override
43. Range
44. Rate
45. Select
46. Sequence
47. Tank
48. Temperature
49. Under
50. Over
51. Valve
52. Voltage
53. Malfunction
54. Stabilization
55. Guidance
56. Life Support
57. Fire
58. Atmosphere
59. Error
60. Experiment event
61. Monitor
62. Interrupt
63. Ready
64. Active
65. Channel
66. Request
67. Data
68. Return to control center
69. Enter data
70. 10 seconds
71. DC power bus failure

Provisions are included in the VTE to permit changing from male voice register to female voice register without changing word length or the 8-bit addressing code. This capability is achieved by a control switch located on the VTE chassis.

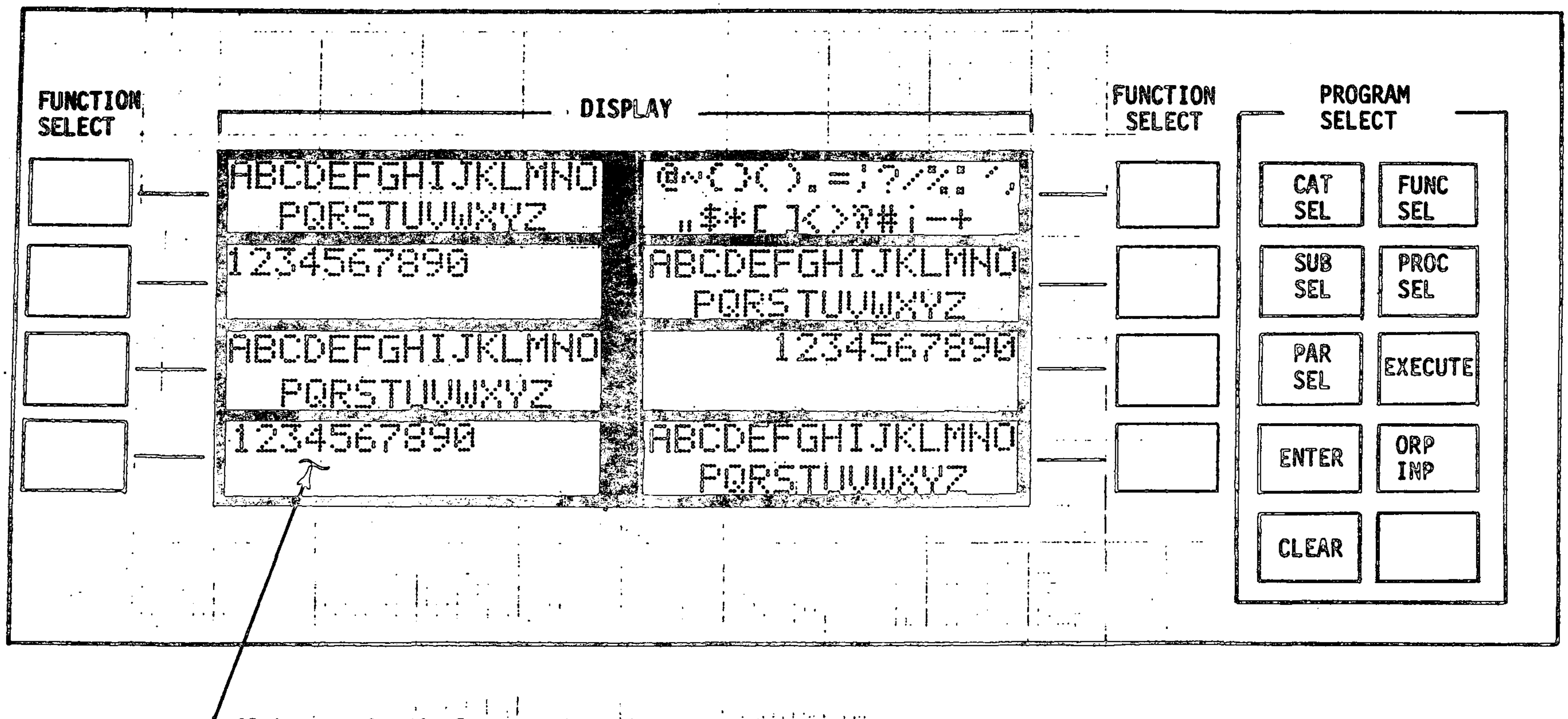
Independent control of speaking rate is provided so that the rate of speech delivery can be varied within the limits of intelligibility without changing pitch. This capability is achieved by use of a chassis mounted control switch. One switch position is for normal synthesized voice and the other position allows manual adjustment capability. Adjustment range is from one-half normal rate to twice normal rate.

Peak output signal from the voice synthesizer is 5 milliwatts (+6 dBm) into a 150-to 600-ohm balanced load. Frequency range of the voice signal is 75 to 5,000 Hz. A loudspeaker is supplied in a separate enclosure to allow flexibility of placement. Capability for mixing an externally generated audio signal into the loudspeaker is also provided. A separate volume control for this external signal is provided to allow independent adjustment. The external audio input signal input will accept a peak input level of 5 milliwatts (+6 dBm) into a 150- to 600-ohm balanced load.

The voice terminal is designed to accommodate increased vocabulary size up to 256 words and additional independent voice synthesizers sharing a common message storage. Expanded vocabulary is provided by addition of modules (single plug-in board), each containing up to 16 plug-in memory cells. cells. Capability is provided for rapidly changing the existing vocabulary to a new vocabulary or new words by changing plug-in module boards or individual plug-in memory cells.

4.2.4 Programmable Function Keyboard

The programable function keyboard is designed and built by MDAC, utilizing a Burroughs Corporation 256-character "self-scan" panel display and pushbutton selection switches. Figure 4-6 shows the PFK configuration, and Table 4-14 lists the major characteristics.



Alphanumeric display for identifying pushbutton switch functions (display shows typical format of characters)

Figure 4-6. Programmable Function Keyboard

Table 4-14

PROGRAMMABLE FUNCTION KEYBOARD MAJOR CHARACTERISTICS

Functions

A. Provides man-machine communications link

B. Dictionary of address codes not required

C. Buildup address codes via sequential selection process

D. Primary computer input device

E. Operational simplicity

F. CRT provides buildup history and status

Features

A. Computer generated variable nomenclature

B. Eight selectable listings

C. Flat panel display panel

D. Function keys for controlling and indicating sequence level

E. Interactive with other input devices.

Display Unit Characteristics

Character Capacity:	256
Text format:	Eight rows, 32 characters per row
Row spacing:	Three blank dot rows
Column separation:	Two blank dot columns
Character form:	5 x 7 dot matrix
Character size:	0.20-in. wide, 0.28 in. high
Center-to-center dot spacing:	0.040 in.
Viewing area:	9.18 in wide, 3.38 in. high
Brightness:	50 ft lamberts
Contrast ratio:	20:1
Color:	Neon red orange
Viewing angle:	100 minimum degrees,
Memory access:	1.75 sec
Scanning rate:	85 Hz.

4.2.4.1 Display Unit

The Burroughs panel display of 256 characters has an array of eight rows with 32 characters per row. This array is subdivided into eight nomenclature groups, each containing two lines of nomenclature with a capacity of 15 characters per line. These eight groups allow for display of up to eight parameters per display frame. Each character is displayed in a 5 by 7 dot matrix. Characters within the same register are separated by two dot columns, and adjacent registers are separated by three dot rows. The panel is 222 dot columns long, with dots located on 0.040-in. center-to-center spacing.

The subsystem is capable of displaying a 64-character repertoire represented by a six-bit code, using a modified USASCII code structure. The unit consists of the following major functional blocks: an MOS character generator, and MOS random access memory (RAM), dot pattern storage buffers and demultiplex circuitry, timing and mode control, and anode and cathode drive electronics required to drive the panel.

4.2.4.2 Pushbutton Selection Switches

Associated with the display are a series of function select and program select illuminated pushbutton switches. The eight function select switches correspond to the eight programmable nomenclature groups and allow the operator to pick one parameter from the group of eight displayed possibilities. The program select switches allow the operator to initiate the selection sequence, advance or backup in the selection process, add additional input data from the A/N keyboard, and control overall PFK operation. The actual functions of the program select switches are controlled by the central DMS computer software and may be tailored to fit various operating concepts. The following is the suggested implementation, based upon results of the Astronaut Computer Communication Study (NAS 8-2501), performed for MSFC by MDAC.

A. Category Select—This switch initiates PFK operation by causing the computer to display the top level listing of functions, from which the operator may select by pressing the appropriate display select switch. (If the category or any lower level listing contains more than eight entries, the second and subsequent activations of the Category Select switch will call up the remainder of the list in groups of eight.)

B. Function Select
C. Subfunction Select
D. Procedure Select
E. Parameter Select

These switches allow the operator to advance or back up through the selection sequence as desired. Each causes progressively lower levels of options to be displayed.

F. Operator Input—This switch conditions the computer to accept additional data via one of the alphanumeric keyboards.

G. Enter—Operation of this switch notifies the computer to read data entered via the alphanumeric keyboard.

H. Execute—This switch causes the computer to execute an instruction that has been "built up" through successive keyboard operations.

I. Clear—This switch is used to "erase" previously entered but unexecuted instructions, such as to clear errors.

4.2.4.3 PFK Operation

Typical use of the PFK as an input device in instructing the computer to perform a function is described in this section. Parameters, such as supporting software, discrete levels of selection capability, and number of displayable groups, are intended to be evaluated during the course of developing the PFK concept, using the display and control breadboard. Usage of the device may indicate changes are required from the baseline approach described below.

Figure 4-7 shows pictorially a hypothetical case where the operator wishes to align a telescope in the intermediate ultra-violet experiment package to a new set of target coordinates.

Initialization of the PFK is accomplished by pressing the Category Select pushbutton, which requests from the host computer the top-level listing of mission functions; i.e., mission control, flight control, subsystems, and experiment program. The operator will select from this listing the

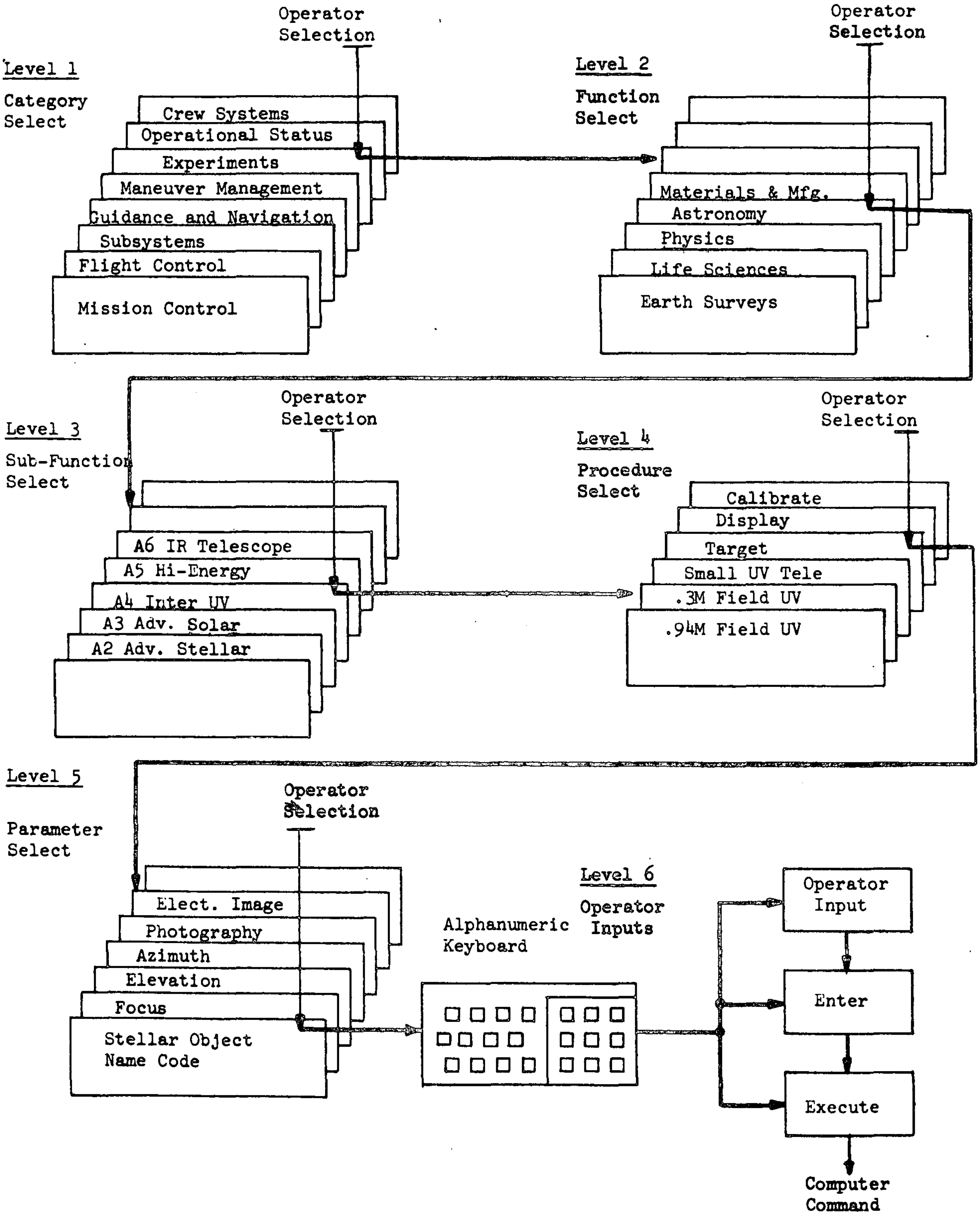


Figure 4-7. Typical Programmable Function Keyboard Operation

appropriate group (in this case experiment program), and the computer will provide a new display listing of options at the next lower level. Associated with this selection (and each subsequent selection), the selected level pushbutton will become illuminated to advise the operator of the level at which he is working. Also, a CRT display may be used in conjunction with the selection process to indicate level and actual selected choice to provide cues for the operator and to allow him to verify proper computer interpretation of his inputs.

The second listing will list the major experiment classes, i.e., astronomy, physics, communications and navigation, technology, life sciences, earth surveys, and material sciences. Selecting the astronomy category identifies to the host computer the next lower level program required and the display associated with this level will be displayed. In a similar manner the operator make selections to select the desired experiment (intermediate ultra-violet) and procedure (point) and with the A/N keyboard types in the new elevation and azimuth coordinate. Depressing the Enter and Execute pushbuttons will activate the computer to direct the telescope to these new coordinates.

Although obviously less direct than a dedicated pointing control for the experiment, such a control method does permit use of a relatively simple device to perform a wide variety of functions, thereby reducing substantially the volume and complexity of the total display and control equipment and providing a high degree of capability to accommodate changes.

4.2.5 Dedicated Displays and Controls

Dedicated display and control devices are supplied that interface with the computer and data bus for control and monitoring of specific functions. These dedicated devices include:

A. Light emitting diode (LED) clock and timer

B. Status light array

C. Panel meters

D. Toggle switches

4.2.5.1 Clock and Timer

A digital calendar clock and a digital timer manufactured by Electronic Research Corporation were selected because they have LED digital readouts. These units were the only functional devices available at the time that used state-of-art display techniques.

The digital calendar clock is a general purpose system clock for displaying real-time accumulation for a full year time period. The display reads out day-of-year on a 1 to 365 numerical basis plus time of day in a 23-hour, 59-minute, 59-second manner. A front panel switch is provided to select 365 to 366 days per year, and the appropriate position may be selected at any time prior to the end of that year. The clock utilizes the 60-hz power line line frequency as the count time base. Front-panel controls provide for presetting the digits by a SET/RUN switch. Placing the switch to the SET position resets the seconds digits to zero and enables the pushbutton switches associated with each of the other display digits. Depressing the pushbutton switches allows the selected digit to advance at a rate of 1 count/sec until the desired number is attained. Returning the switch to the RUN position starts the clock counting from the preset state. External inputs for run/hold and reset to zero are provided through the data bus interface. These commands are effected by providing external inputs to logic "0" (ground). Placing the run/hold input to logic "0" forces the clock into a hold status with the readout remaining at its present position. Releasing the input to logic "1" enables the clock to continue counting, thus providing for clock synchronization with other standards or clocks. A 1-pulse/sec output is provided from the clock for use as the input count source for the digital timer.

The digital timer is a four-digit accumulator that can count up or count down from a preset condition. The reverse counting sequence when counting down through zero is 003, 002, 001, 000, -001, -002, -003, etc. Computer-controlled presetting of the count may be accomplished by a BCD code via the data bus interface. Front-panel pushbutton switches for start, stop, and reset are provided for manual control. Remote start, stop, reset, and count direction capability is provided by the data bus interface. The timer will count at a 1/sec rate from the signal source provided by the calendar clock.

4.2.5.2 Status Lights

A status light assembly of the type used on the Skylab Program (MDAC Specification No. 1B69937) will be supplied. This assembly consists of ten individual annunciators combined into an array. The individual legends for each annunciator are replaceable at field level. The annunciators interface with the data bus via lamp driver circuits and logic in the display interface adapter.

4.2.5.3 Dedicated Meters

Four meters will be provided on the display and control breadboard to display specific dedicated analog parameters. These units interface with the data bus through the display interface adapter (DIA), which provides digital to analog conversion and drive electronics for the meter movements. The meters are built by International Instruments, Inc., and have replaceable meter scale inserts so that scales can be assigned or changed at a later date when assigned parameters are known. The labeling for nomenclature on these panel meters are also changeable and replaceable in the field.

4.2.5.4 Dedicated Controls

Eight panel-mounted toggle switches are provided to simulate Space Station dedicated controls. These switches interface with the data bus through the DIA, which encodes the contact closures and assigns the proper address code to identify the switch that is activated. Both maintained and momentary contact switches will be provided to simulate typical subsystem discrete inputs and commands.

4.2.6 Display Interface Adapter

The display interface adapter (DIA) supplied by IBM provides the interface between the data bus terminal (DBT) and the console control and display equipment. The unit receives, serial data and a clock signal at 1 megabit per second from the DBT and converts the data to parallel form for input to the display devices. In the reverse direction, the DIA receives data in parallel form from the console equipment and converts it to serial form for output to the DBT.

Data inputs from the DBT are in the form of a command word (data bus "B" word), followed in some cases by up to 32 data words of 16 bits (plus a lead bit and a parity bit) each. The DIA decodes the command word to determine which console device the command is addressed to and to determine the requested function (i.e., read or write), and it then proceeds to execute the command. The DIA logic provides the "hand-shaking" necessary to verify that the proper conditions for transfer exist, and it provides buffering where required to achieve compatibility between the DBT input/output rate and the console device data rate. The DIA also detects certain classes of errors in the data, such as format and parity errors. Flag bits indicating the detection of these errors are stored in an internal status register along with other status information, and may be read out via the data bus under computer control.

A block diagram of the DIA logic is shown in Figure 4-8. A detailed design specification for the unit is contained in Appendix D of this document.

4.2.7 Self-Test Features

The display and control console contains several self-test features that are designed to simplify functional testing and troubleshooting of the assembly. These are described in the following sections.

4.2.7.1 Lamp Test

A Lamp Test switch is provided on the front panel. Activation of this switch energizes the lamp driver circuits in the DIA and lights all indicator lamps and illuminated pushbutton switches simultaneously.

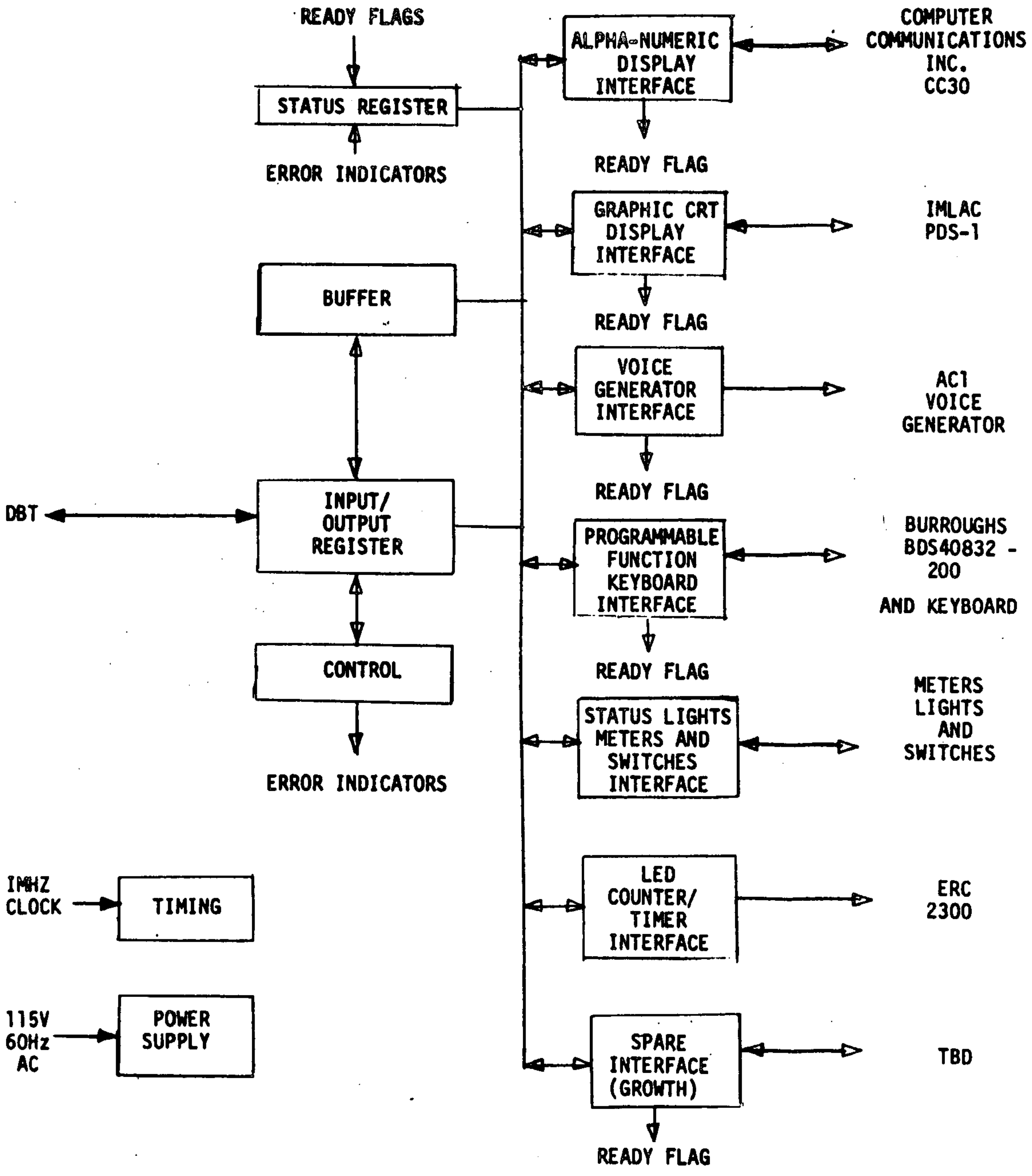


Figure 4-8. Display Interface Adapter Block Diagram

4.2.7.2 Voice Message Generator Self-Test

The voice message generator unit has the capability to sequentially step through its vocabulary upon activation of a test switch. In the event a message code is received while the unit is cycling through the vocabulary, a priority interrupt condition shall cause the test cycle to stop, permitting the required message to be generated. After completion of this required message, manual activation is required for re-initiation of the test cycle. The test switch shall be a multiposition switch. One switch position causes the unit to automatically sequence through the complete vocabulary, and the other permits manually stepping through the vocabulary one word at a time.

4.2.7.3 Alphanumeric Display Self Test

The CC-301 display controller is equipped with a Test/Normal switch, which when in the Test position, causes test data patterns to appear on the display screen. These consist of a unique pattern of alphanumeric characters if the manual Graph/Alpha switch is in the Alpha position, and a pattern of dots if the switch is in the Graph position.

4.2.7.4 PDS-1 Self-Test

The PDS-1 keyboard, processor logic, and CRT can readily be tested by simple keyboard operations that exercise the machine through random or preplanned routines. Various standard diagnostic and demonstration software programs are also available, including memory test routines.

4.2.7.5 Console Self-Test

Provision has been made in the DIA to allow the PDS-1 minicomputer to serve as a test data generator for exercising and testing the other console displays. Through appropriate programming of the PDS-1 computer, test messages can 'be output to perform the following functions.

A. Individually light each of the discrete indicator lamps and lighted push-button switches.

B. Provide varied outputs to drive the analog meters.

C. Output test messages to the alphanumeric CRT.

D. Display test messages on the alphanumeric portion of the programmable function keyboard.

E. Generate voice message outputs from the voice synthesizer.

F. Start and stop the LED clock display.

G. Preset, start, stop, and reset the LED event-timer display.

In addition, it is possible to create typical display presentations on the PDS-1 CRT, including both alphanumeric and graphic information and providing capability for testing the light pen.

Section 5
COMMUNICATIONS

This section describes the Ku-band communications equipment, which is deliverable under the IMS special emphasis task. The equipment includes an exciter, developed by Collins Radio, and a power amplifier manufactured by Hughes Aircraft Company.

5.1 K$_U$-BAND EXCITER

The K$_u$-band exciter is a completely solid-state unit, operating at a frequency of 14.500 GHz, at an output power level continuously adjustable between 5 and 50 Mw. The unit is constructed to permit mounting in a standard 19-inch relay rack.

5.1.1 System Description

The exciter consists of a modulation amplifier, a 2 GHz FM oscillator, a 2 GHz amplifier, an AFC unit, an AFC reference unit, a test coupler, three frequency doublers, a transmit filter, and load isolators.

Five separate modulation inputs are provided with level controls on each input. These inputs consist of TV video (41.25 to 47.25 MHz), digital modem data (up to 10 Mbps), experiment data (up to 10 Mbps), baseband video data (up to 10 MHz), and a voice input (200 to 3500 Hz).

5.1.2 Principles of Operation

The following circuit description covers the K$_u$-Band exciter. Detail circuit descriptions of the individual modules are provided in the module information section.

5.1.2.1 Transmitter

Refer to the simplified block diagram in Figure 5-1 for the following discussion of the exciter.

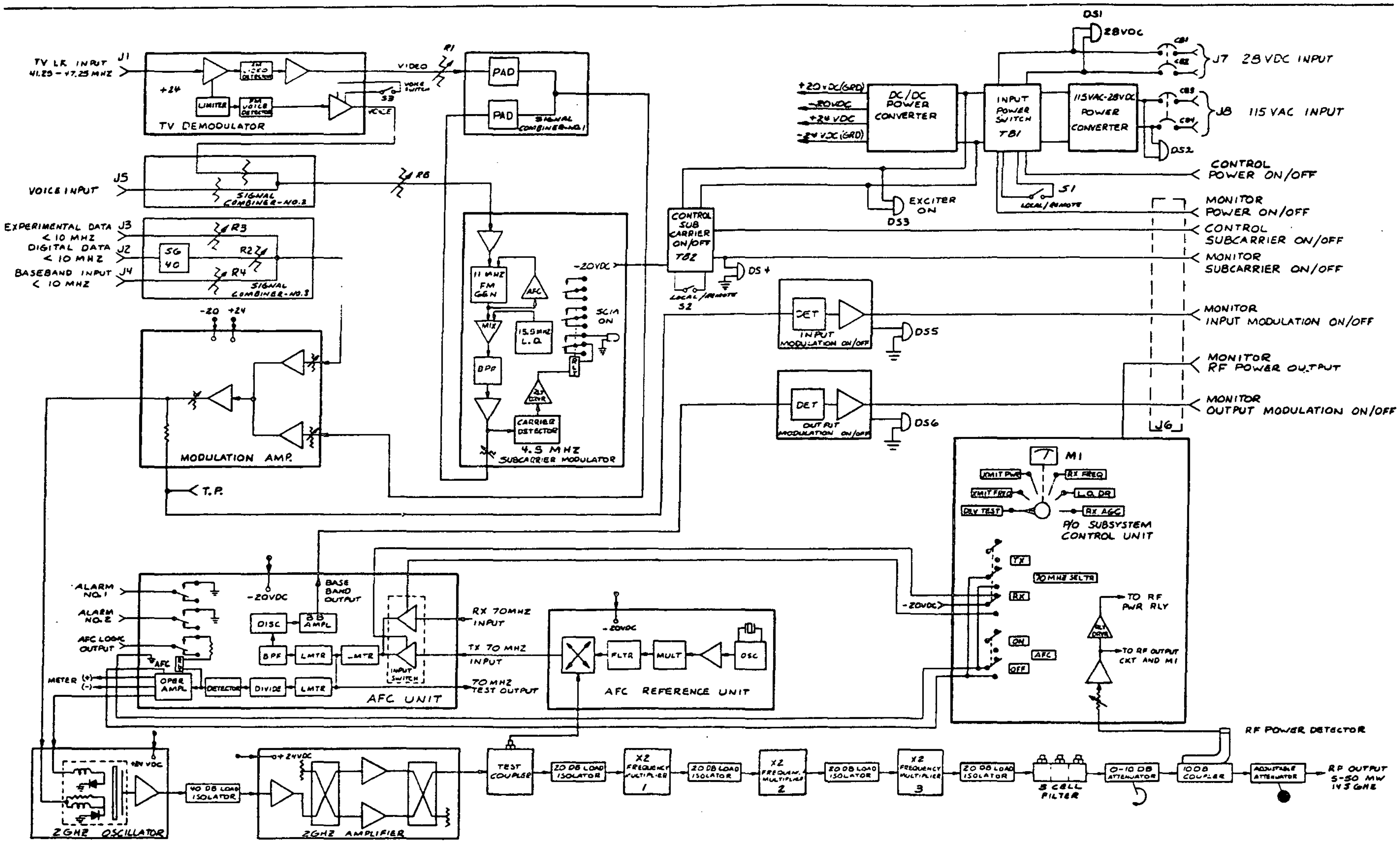


Figure 5-1. Block Diagram – K_u Band Exciter

5.1.2.2 Modulation Amplifier

The modulation amplifier amplifies the baseband input signal to the drive level necessary to modulate the microwave oscillator. Signals are applied through the control panel to the inputs of the broadband microwave modulation amplifier. The signals contain voice, data, or video information.

The two isolated baseband inputs are combined in an emitter follower and connected to a three-stage output amplifier. An output level control sets the gain of the amplifier, and an isolated (30 dB) test jack is provided to monitor the output level.

5.1.2.3 2-GHz Oscillator

The 2 GHz oscillator generates the 2 GHz fundamental frequency of the microwave transmitter and uses only one active element, a coaxial transistor driving the center conductor of a coaxial cavity resonator. This FM signal is amplified in power by the power amplifier and multiplied up to the assigned transmitter frequency by the three frequency doublers. The baseband signal from the modulation amplifier is capacitively coupled to the oscillator by a varactor diode probe, which permits linear deviation of the oscillator to produce an FM output signal from the osicllator. The cavity oscillator with AFC has a frequency stability of 0.0005 percent or 42 KHz from -30° to 50° C.

5.1.2.4 Load Isolators

The 40 dB load isolator is a coaxial ferrite device that isolates the cavity oscillator from the effects of mismatching in the input impedance of the power amplifier. The isolator has a forward attenuation of less than 0.5 dB and a reverse attenuation of 40 dB. The 20 dB load isolators are similar except the reverse attenuation is 20 dB.

5.1.2.5 Power Amplifier

The power amplifier raises the power level of the modulated oscillator signal enough to overcome the component insertion losses and produce an output of the transmitter subsystem of approximately 50 mw. There are three active elements in the power amplifier; a transistor driver amplifier and two parallel transistor power amplifiers. The output of the driver amplifier is split by a power divider to drive the parallel transistor amplifiers. The output of the parallel amplifiers is added through a power combiner to produce a nominal 5 watt output from 0.5 watt input.

5.1.2.6 Coaxial Test Coupler

The coaxial test coupler provides a sample of the transmitted power to the afc reference unit and a transmit power test jack. Each connector is isolated 30 dB from the through signal. The sample signal to the AFC reference unit is approximately 5 milliwatts.

5.1.2.7 Frequency Doublers

The frequency doublers are used to multiply (X8) the 2 GHz frequency from the basic oscillator to the desired 14.50 GHz transmitter frequency. The frequency doublers contain an input impedance matching network and a filter to select the desired input frequency, varactor diodes, and an output filter with an impedance matching network. The output filter allows only the desired band of frequencies (second harmonic of the input frequency) to pass through to the output connector.

5.1.2.8 Transition Isolator

The 20 dB transition isolator provides a coaxial to waveguide transition to convert the transmitted signal from a coaxial transmission path to a waveguide transmission path. It also provides a load isolator as a waveguide ferrite device that isolates the doubler from changes or effects of mismatching in the waveguide transmission line. The isolator has a forward attenuation of less than 0.5 dB and a reverse attenuation of 20 dB.

5.1.2.9 Power Detector

The power detector is used to determine the relative transmitted power as indicated on the meter in the subsystem control unit and also provides an RF test jack so that the transmit power can be measured at this point.

5.1.2.10 Transmitter Filter

The transmitter filter is a three-cell, rectangular-cavity filter constructed in a section of waveguide. The cells are separated by circular coupling irises, and a tuning screw is mounted in each cavity section. These tuning screws make the proper tuning of the filter over its frequency range possible. Because the basic filter is designed at a frequency slightly higher than the highest usable frequency, inserting the screws increases the capacity loading, thereby tuning the filter to a lower frequency.

5.1.2.11 AFC System

Automatic frequency control is used to stabilize the FM oscillator's long-term (at rest) frequency while still permitting the oscillator to be deviated by a modulating signal. It protects against temperature variations, component aging, and power-supply-induced changes by comparing the frequency of the transmitted signal to the frequency of a crystal-controlled oscillator. Any difference from this comparison is used as a control voltage to correct the frequency of the fm oscillator. The AFC reference unit and AFC unit provide the automatic frequency control functions for the transmitter.

AFC Reference Unit

The AFC reference unit contains an oscillator and multiplier circuit, which produce a reference frequency that is 70 MHz below the transmitter frequency. The reference frequency is mixed with the sampled transmitter frequency in a microstrip mixer to produce a 70 MHz difference frequency.

AFC Unit

The AFC unit receives the 70 MHz signal from the AFC reference unit. This signal is then limited, discriminated, and supplied to a baseband amplifier. The baseband output is used for linearity adjustments, setting system deviation, modulation presence, and transmitter frequency sensing. The 70 MHz signal is also applied to a 64:1 divider to produce a 1.09375 MHz signal. A crystal-controlled reference oscillator generates a 1.09375 MHz signal. Both 1.09375 MHz signals are supplied to 65000:1 dividers, and the resultant 16.8 Hz signals are connected to a phase detector to develop a dc error voltage. The 1.09375 MHz signals are reduced to 16.8 Hz so that one cycle, with modulation, will not exceed the range of the phase detector.

5.1.2.12 Power Supply

The power supply subsystem consists of two basic units, the 115 vac to 28 vdc power converter and the dc-to-dc power converter. The system is constructed to permit operation from either 115 vac, 60 Hz, single-phase power or a dc voltage of 28 ± 4 vdc (with negative ground). This system provides the operating voltages for the exciter. Two regulated outputs (to 5 percent) supply -20 volts and +24 volts to operate all modules in the exciter.

5.1.2.13 Subsystem Control Unit

The sybsystem control unit is used to monitor, test, and adjust the transmitter/receiver subsystem. The unit provides the following meter functions:

A. Transmitter deviation

B. Transmitter power output (relative)

C. Transmitter frequency (relative)

A toggle switch in the control unit connects the transmitter afc correction voltage to the afc probe of the transmit oscillator. This switch is used to check AFC pull-in and to disconnect the AFC control loop when checking the receiver deviation.

5.1.2.14 Patch Panel (Jackfield)

The jackfield is used to connect signals into and out of the subsystems. BNC coaxial connectors provide input (Xmt 1 and Xmt 2) connections. Each input and output has a 30 dB isolated test point (BNC) for maintenance and testing. The transmitter AFC baseband output (XMT SNSR), which is used for setting the transmitter deviation, is also connected to the jackfield.

5.1.2.15 Subcarrier Modulator

The subcarrier modulator operates at an output frequency of 4.5 MHz. This subcarrier is frequency-modulated by the external voice or the internal TV voice signal. The frequency modulated 4.5 MHz subcarrier frequency deviates the 14.500 GHz carrier.

5.1.2.16 TV Demodulator

This unit demodulates the TV IF signal (41.25 to 47.25 MHz) into a video and an audio signal. The audio is used to frequency deviate the 4.5 MHz subcarrier, which is then combined with the video. The combined signals are then used to frequency deviate the main carrier.

5.1.2.17 Control and Indicator Panel

This assembly contains input and output connections and controls, which are pertinent to the operation of the system. Individual inputs are provided for each of five modulation signals, and controls (attenuators) are provided for each input. Controls and connections are provided to turn the unit on, either locally or remotely, and to completely disconnect it from the power line. Indicators are provided to determine the status of the controls and the unit.

5.1.3 Performance Characteristics

5.1.3.1 Electrical Performance

Operating Frequency

The exciter operates at a nominal frequency of 14.5 GHz with an instability of not more than ±0.01 percent from the assigned frequency.

RF Power Output

RF output power is not less than 5 or not more than 50 milliwatts into a 50 ohm load with a VSWR of 2 to 1 (or less) at any phase angle. The power output is continuously adjustable over the above range.

Output Impedance

Nominal output impedance of the exciter is 50 ohms.

Carrier Deviation

The exciter provides a frequency-modulated carrier having the following peak deviations for the signals listed under Signal Inputs.

A. Television—5 MHz ± 10 percent

B. Baseband—5 MHz ± 10 percent

C. Digital modem data—4 MHz ± 10 percent

D. Experiment data—4 MHz ± 10 percent

E. An internally generated 4.5 MHz subcarrier frequency frequency-deviates the carrier 400 kHz ± 10 percent peak.

Signal Inputs

The exciter is capable of accepting the following signal inputs and of providing for the carrier deviations specified under Carrier Deviation.

A. A commercial television receiver IF signal (41.25 to 47.25 MHz). The minimum input level will be 0.5 volt peak. The input impedance of this port is nominally 75 ohms.

B. A baseband containing frequencies up to 10.0 MHz. The minimum input level is 0.5 volt peak. The nominal input impedance of this port is 75 ohms.

C. A serial digital data stream at rates up to 10 Mbps. The required input logic conditions are as follows:

Logic "1" - Greater than 2.4 volts at 0.5 ma load current to ground or 120 microamperes to a positive supply.

Logic "0" - Less than 0.4 volt with a sink of 5 ma (or lower) load current.

D. A data stream of experimental data at rates up to 10 Mbps. The nominal input impedance is 75 ohms. Logic levels are as follows:

Logic "1" - 5 ± 0.5 volts

Logic "0" - 0 ± 0.4 volt

E. An audio signal containing frequencies between 200 and 3500 Hz shall frequency deviate the 4.5 MHz subcarrier 20 kHz ± 10 percent peak. The required audio input level is 0 dbm. The nominal input impedance is 600 ohms balanced.

Linearity

The exciter has a nonlinearity of NMT ± 1.0 percent for sine-wave modulation inputs up to 10 MHz and deviations up to 16 MHz peak.

Control Inputs

The exciter is capable of being controlled by the controls listed below:

A. Primary power - ON/Off

B. 4.5 MHz subcarrier - ON/Off

The control input signal has the following characteristics:

A. "On" state - 5 ± 0.5 vdc into a 1000-ohm load.

B. "Off" state - 0 ± 0.4 vdc.

C. Risetime - Less than, or equal to, 1 millisecond.

D. Short-circuit current limited to 10 ma maximum.

Monitor Outputs

The exciter provides the following monitor outputs to a remote data acquisition unit (RDAU). Digital voltages are 0 or 5 volts and analog voltages will be 0 to 5 volts, normalized. The nominal load impedance for these outputs is 1000 ohms.

A. Primary power - ON/OFF (bilevel)

B. RF power output - (analog)

C. Modulation input present - (bilevel)

D. Modulation output - present - (bilevel)

E. 4.5-MHz subcarrier - ON/OFF (bilevel)

Spurious Response

Spurious frequencies and harmonics above 1 GHz are attenuated at least 60 db, and those below 1 GHz are attenuated at least 50 db.

Duty Cycle

The exciter is capable of continuous operation.

Input Power

The exciter may be operated from either 115 volt, 60 Hz, single-phase primary power or +28 vdc primary power. Indicators on the front panel indicate which power source is present. The unit requires NMT 100 watts.

5.1.3.2 Mechanical Design and Construction

The mechanical design and construction is as outlined in the following paragraphs:

Packaging

The exciter is packaged as a single unit in a configuration that permits installation in a standard 19-inch relay rack.

Control, Indicator, and Connector Locations

The following controls, indicators, and connectors are provided on the exterior of the unit.

A. A master power switch for either the 28 vdc or 115 vac primary power. A visual indication is present, which indicates when and which primary power is present.

B. A pushbutton exciter power ON/OFF switch and a visual indication on the front panel. The unit may be turned on locally or by remote control.

C. A push-button 4.5 MHz subcarrier ON/OFF switch with a visual indication on the front panel. The subcarrier may be controlled locally or by remote control.

D. The RF output connector is located at the rear of the unit. The output connector is WR62 (RG 91/U) waveguide with a cover flange.

E. Separate connectors are provided on the front panel for each of the signal inputs. All of these except the voice-input connector are coaxial BNC. The external voice input connector is a Cannon Electric XLR-3-31, 3 contact socket.

F. Visual indicators, located on the front panel, are provided for the presence of modulation at both input and output, and the status of the 4.5 MHz subcarrier.

G. A variable attenuator with 20 dB minimum range is provided on the front panel for each of the signal inputs.

H. The power connectors for 28 vdc or 115 vac are located at the rear of the unit.

I. Control inputs and monitors output are located in one connector at the rear of the unit. This connector is a Bendix-type PT02A-16-26P connector with 26 pins.

Thermal Dissipation

The exciter operates without the necessity for external coolant, in an ambient of +75 ± 25° F.

5.1.4 Detailed Subassembly Descriptions

The electrical characteristics and a description of the function and operation of the subassemblies that compose the K_u-Band exciter are described in this section.

5.1.4.1 Power Supply Subassembly

Purpose of Subassembly

The power supply subassembly consists of two basic units: (1) a 115 vac, single-phase, 60 Hz to 28 vdc converter, and (2) a dc-to-dc converter. This subsystem is used to provide the -20 and +24 volts from either a 115 vac source or a 28 vdc source (negative ground).

Subassembly Description

115-VAC to 28-VDC Power Converter—This unit is a Lambda model LM-CC-28 solid state, modular supply. The unit is 4-15/16 by 4-15/16 by 9-3/8 inches and is mounted in the control and indicator assembly at the bottom of the exciter. The electrical characteristics are as follows:

Fixed voltage range (vdc)	28 ±5 percent
Maximum amps at 40° C	3.5
71° C	2.8
Ripple (mV RMS)	1.0

DC-to-DC Power Converter—The unit is a transistorized, etched-circuit module with a sheet-metal cover, and is mounted within the RF subsystem cabinet. Unit dimensions are 7 by 10-1/2 by 1-3/4 inches. The electrical characteristics are listed in Table 5-1.

Table 5-1

POWER CONVERTER CHARACTERISTICS

Input voltage

20 to 56 dc (either pin can be connected to chassis ground)

Input ripple:

250 millivolts, maximum

Output voltage and current:

20 volts dc at 2 amperes

24 volts dc at 2 amperes

(Either output can be grounded to meet system requirements.)

Regulation:

±2 percent, no load to 0.05 ampere

±5 percent, 0.05 ampere to full load

Output ripple and spikes:

Less than 10 millivolts

Short circuit protection:

Either output can be shorted for 30 seconds (automatic recovery).

Undervoltage protection:

Unit shuts off when source drops below 18 volts (automatic recovery).

Overvoltage protection:

Unit shuts off if 24 volt output exceeds 29 volts, or if 20 volt output exceed 24 volts. (Source voltage must be removed for recovery.)

Circuit Description

The power supply subsystem in this unit provides for operation from either a 115 vac source or a dc input at any voltage between 20 and 56 volts. The system requires that the minus (-) dc voltage be system (chassis) ground.

The 115 vac power input is converted to 28 vdc by the 115 vac to 28 vdc power converter. The output is used to feed the dc-to-dc converter. If a dc voltage is used as a primary power source, it is fed directly to the dc-to-dc converter through blocking diodes.

The dc-to-dc converter is used to supply operating voltages for a transmitter subsystem. The unit operates from any input between 20 and 56 vdc (without strapping) and provides outputs of 20 and 24 dc.

The input dc voltage is converted in the power supply to an alternating voltage that is rectified and filtered to furnish two dc output voltages. The average dc level of the alternating voltage is controlled, so regardless of input voltage level (within operating limits), the output voltages remain constant.

5.1.4.2 2 GHz Amplifier Subassembly

Purpose of Subassembly

The 2 GHz amplifier is used with a microwave transmitter to raise the FM oscillator output to a nominal 5 watt level. An external voltage-adjust assembly controls the amplifier output. The output is normally monitored at the test coupler during in-service power and frequency checks.

Subassembly Description

The unit is a solid-state assembly in a hermetically sealed aluminum box. The unit is mounted within the RF subsystem cabinet. Unit dimensions are 4 by 5-1/2 by 1 inches. The electrical characteristics are as follows:

Power requirements:
+24.0 ±0.5 volts at 1.4 amperes, maximum
Input and Output Impedance:
50 ohms
Output power
5 watts nominal
Power Gain:
9.25 dB minimum
Frequency range:
1606 to 2001 MHz.

Circuit Description

The 2 GHz amplifier is a power amplifier that raises the modulated oscillator signal to the level required to drive transmit output circuits or frequency doublers. The unit is a hermetically sealed integrated circuit and operates as a Class C amplifier with a nominal output of 5 watts. This subassembly consists of two major circuits. The driver circuit consists of an input filter, an amplifier, and an output filter. The input modulated signal is amplified, filtered to suppress harmonics, and applied to the final amplifier subassembly. This portion of the power amplifier contains a power divider, twin Class C amplifiers with input and output filters, and a power combiner. The amplified and filtered signal from the driver subassembly is applied through the power divider to the parallel filter and amplifier circuits. The outputs of these two circuits are combined in the combiner network and coupled to the transmit coupler. At the coupler, a sample of the signal is diverted to the AFC reference unit.

5.1.4.3 AFC Reference Subassembly

Purpose of Subassembly

The AFC reference unit samples the output of the 2 GHz amplifier and provides this output as an IF signal to the AFC unit for transmit frequency control.

Subassembly Description

The unit is a transistorized, microstrip, etched-circuit card contained in a chromate-coated aluminum box. The unit is mounted at the top of the main assembly. Unit dimensions are 3-1/4 by 3-3/4 by 6-7/8 inches. The electrical characteristics are as follows:

Power requirements

-20 volts at 140 ma, nominal (585 ma maximum at turn-on)

Transmitter input signal level

+7 dBm nominal

Output signal level

-25 to -45 dBm nominal

Circuit Description

The AFC reference unit mixes a sample of the 2 GHz amplifier output signal with a crystal-controlled reference signal to produce a nominal 70 MHz difference signal. This difference signal is applied to the AFC unit and used in a phase-comparison loop to hold the transmitter on frequency.

The unit consists of a reference frequency generator composed of an oscillator, a doubler, and amplifiers; a multiplier network composed of a strip-line, low-pass filter, step-recovery diode (SRD), and a resonant cavity; and a mixer/filter network containing a 2.3 GHz low-pass filter, a 10 dB coupler, a mixer diode, and a 70 MHz output filter.

The oscillator circuit products a stable signal at the required frequency, and the oscillator crystal is over-contained for temperature control. The oscillator output is frequency doubled, amplified, and applied to the multiplier network. SRD, in conjunction with the multipler input network, produces a signal rich in harmonics, which is coupled to the tunable cavity. The cavity is tuned to select the desired harmonic.

The selected harmonic is applied through a low-pass filter and a 10 dB coupler to a diode, where it is mixed with the input sample of the 2 GHz amplifier output signal. Of the frequencies that result from the mixer action, only the nominal 70 MHz difference frequency is passed by the output filter section. This output signal is routed to the AFC unit and used in transmit frequency control.

5.1.4.4 AFC

Purpose of Subassembly

The AFC unit provides transmit frequency control.

Subassembly Description

The unit consists of two transistorized, etched-circuit cards in an aluminum shield box with a chromate-coated cover. The unit is mounted on the inside of the RF subsystem cabinet door. Unit dimensions are 5 by 12 by 2 inches. The electrical characteristics are as follows:

Power requirements:	- 20 volts
Alarm outputs for excessive AFC:	- 20 volts
Input impedance:	75 ohms, dual inputs

Circuit Description

The AFC unit demodulates a nominal 70 MHz input signal from the transmitter AFC reference unit. The dc component of the demodulated transmit signal is used to correct any frequency drift of the transmitter FM oscillator, and the ac component (baseband) is used to provide a deviation indication. The AFC unit consists of a 70 MHz IF amplifier card and a 70 MHz AFC card.

The 70 MHz IF amplifier contains a voltage regulator, an input switch, a squelch circuit, and a discriminator. The 70 MHz signal is amplified and then rectified by a voltage-doubling detector. The output of the detector drives a squelch amplifier. If the input signal drops to the squelch level (-50 ±5 dBm), a negative voltage increase turns off the amplifiers. The normal function of these amplifiers is resumed when a signal above the squelch level is applied to the input. The 70 MHz signal is then amplified, filtered, and fed to the 70 MHz AFC card.

The 70 MHz AFC circuit consists of a voltage regulator, frequency dividers, a phase detector, a crystal oscillator, and an AFC sidabled circuit.

The 70 MHz signal from the IF amplifier portion of the AFC unit is applied through to two frequency dividers, which divide the signal down to 16.7 Hz. The 16.7 Hz signal is then applied to a phase detector.

The phase detector compares the 16.7 Hz signal to a reference oscillator signal to produce an AFC correction voltage. The reference signal is supplied by a crystal-controlled oscillator. The 1.09375 MHz output of the oscillator is reduced to 16.7 Hz by cascade divide-by-16 circuits. The two signals applied to the phase detector produce a 16.7 Hz square-wave output.

The duty cycle of the square wave varies with the relative phase angle between the two inputs. The square wave is amplified and filtered and then applied to the FM oscillator to correct any variation in transmitter output frequency.

If the 70 MHz input signal is lost, the loss-of-input detection circuit causes the AFC correction voltage to go to its nominal or pedestal value (-10 volts). With the pedestal voltage applied, the FM oscillator is tuned so that the oscillator rest frequency approximates the required operating frequency. Thus, if AFC control is lost, the transmit frequency remains at or near the desired frequency.

5.1.4.5 Modulation Amplifier Subassembly

Purpose of Assembly

The modulation amplifier is used to amplify the baseband input signal and to provide the required modulation signal level to an FM oscillator.

Subassembly Description

The unit is a transistorized, etched-circuit card contained in an aluminum-shielded box. The unit is mounted within the RF subsystem cabinet. Unit dimensions are 6 by 5-11/32 by 1-7/8 inches. The electrical characteristics are as follows:

Power requirements:
-20 volts at 53 ma
+24 volts at 62 ma

Input impedance
75 ohms, dual inputs

Output impedance
Approximately 2000 ohms

Gain:
38 dB+1, -2dB

Frequency response:
±0.1 dB from 10 Hz to 10 MHz

Circuit Description

The modulation amplifier amplifies the input baseband signal(s) to the level required to modulate the transmitter FM oscillator. The amplifier consists of two three-stage input amplifier circuits, an isolation amplifier and a three-stage output amplifier.

5.1.4.6 2 GHz FM

Purpose of Subassembly

The 2 GHz FM Oscillator is used in the transmitter to generate a carrier in the 2 GHz range and to modulate the carrier with a baseband signal. The unit covers a frequency range of 1781 to 1938 MHz.

Subassembly Description

Each unit is a solid-state device contained in a machined brass housing that is silver-plated with gold flashing. The unit is mounted within the RF subsystem cabinet. The electrical characteristics are as follows:

Power requirements:

+24 volts at 150 ma, maximum
-20 volts at 0.05 ma, maximum

Output power:

500 mw, minimum

Output impedance:

Approximately 50 ohms

Output frequency:

1781 to 1938 MHz

Circuit Description

The 2 GHz FM oscillator generates a 2 GHz carrier and modulates a carrier with the baseband signal from the modulation amplifier. The unit consists of an oscillator circuit, an AFC probe assembly, and a modulation probe assembly. A transistor and its associated components form a tuned-base, tuned-collector oscillator. The tuned-base circuit is developed by a combination of conventional components, microstrip circuits, and construction capacitance. The tuned-collector circuit is a coaxial resonant cavity that contains the AFC and modulation probe assemblies. The AFC probe assembly contains an inductor and a varactor diode. Any change in AFC voltage varies the interelectrode capacitance of the diode, thereby changing the resonant frequency of the cavity. The modulation probe assembly also contains a varactor diode. The modulating signal applied to the unit is coupled to the modulation probe through a parasitic suppressor. The signal varies the interelectrode capacitance of the diode, and this frequency modulates the oscillator output. An RF output probe couples the modulated energy from the cavity, and this energy is applied to the transmitter power amplifier.

A cavity adjustment varies the oscillator frequency and a power output control varies the output power from zero to approximately 600 milliwatts. The three probe assemblies, power, AFC, and modulation, are adjusted for the correct power transfer, modulation sensitivity, and linearity respectively.

5.1.4.7 Control and Indicator Subassembly

Purpose of Subassembly

The control and indicator unit contains the electronics necessary to control the exciter and indicate the status of the system. The unit receives, conditions, and combines all modulation inputs prior to frequency modulation of the exciter. The control and indicator unit also provides for control of primary power, exciter power, and level of each modulation input.

Subassembl Description

The control and indicator assembly is constructed to permit mounting in a standard 19-inch relay rack. The height of the unit is 7 inches. The entire assembly is permanently attached to the bottom of the exciter assembly. Unlatching of the Dzus fasteners permits the front panel to swing down. The panel is hinged at the bottom.

This unit contains the following:

A. 28 vdc circuit breakers.

B. 115 vac circuit breakers.

C. 115 vac to 28 vdc power converter.

D. TB1-input power switch control.

E. TB2-subcarrier modulator switch control.

F. Signal combiners - Number 1, Number 2, and Number 3.

G. 4.5 MHz subcarrier modulator.

H. TV demodulator.

I. Modulation presence detector.

J. The front panel contains the indicators and controls as shown in Figure 5-2.

The electrical characteristics are as follows:

A. Input power requirements

1. 115 vac ± 10 percent, 55 to 65 Hz, single-phase at NMT 100 watts (J8).

2. 28 ± 4 vdc, negative ground; input ripple 250 MV maximum, 65 watts maximum (J7).

B. Modulation inputs

1. Digital data (Modem) input (J2).

a. Logic "1" - greater than 2.4 volt at 0.5-ma load current.

b. Logic "0" - less than 0.4 volt with sink of 5 ma or less load current.

c. Frequency - not more than 10 MHz.

2. Experiment data input (J3)

a. Logic "1" - +5 V ± 0.5 volt.

b. Logic "0" - 0 ± 0.4 volt.

c. Input impedance - 75 ohms matched.

d. Frequency - not more than 10 MHz.

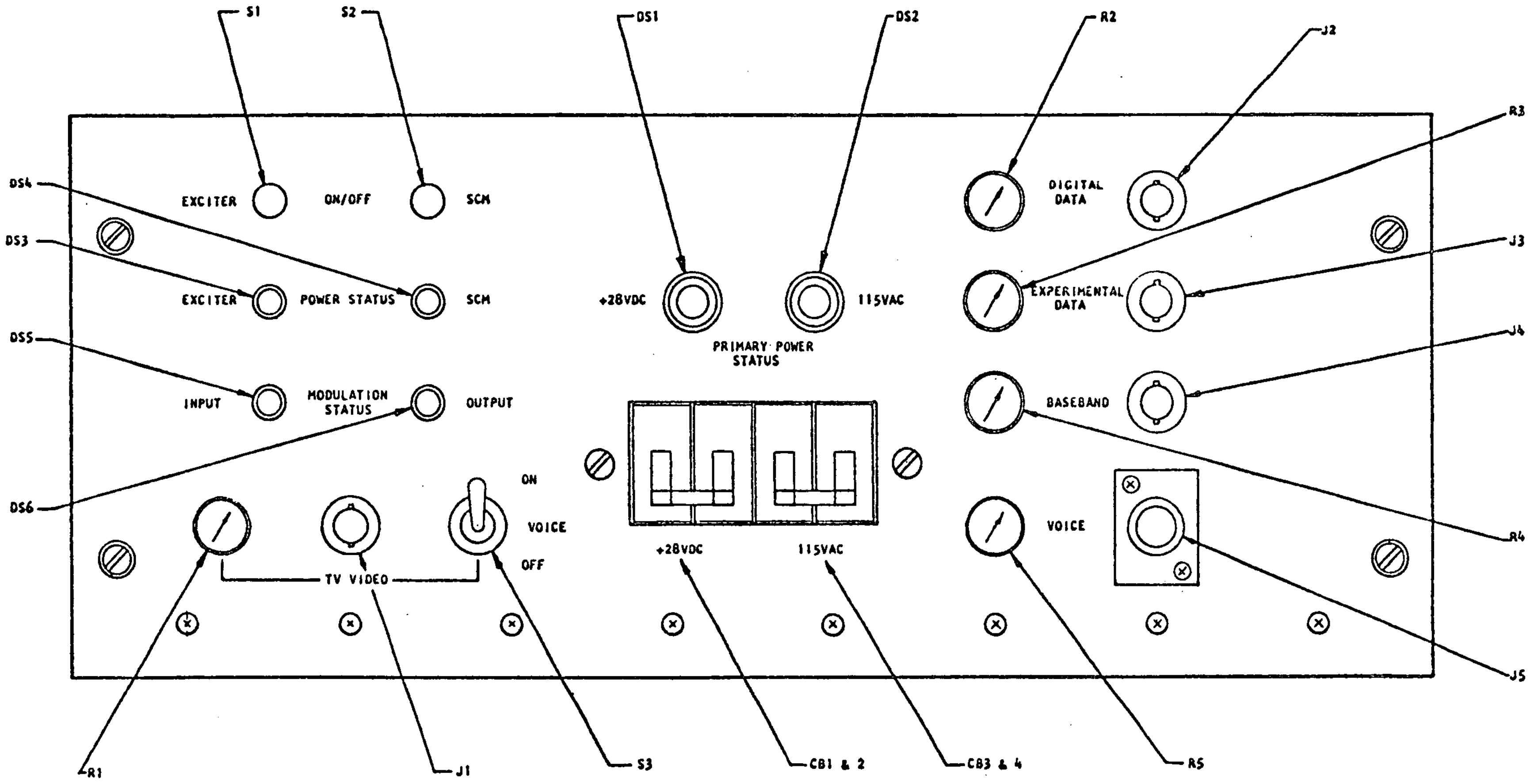


Figure 5-2. Front Panel K_u Band Exciter

3. Baseband input (J4)

   a. Level - 0.5 volt peak (nominal).

   b. Input impedance - 75 ohms (nominal).

   c. Frequency - sine wave, less than 10 MHz.

4. Voice input (J5)

   a. Level - 0 dbm (nominal)

   b. Frequency - 200 to 3500 Hz.

   c. Input impedance - 600 ohms balanced (nominal).

5. TV signal input (J1)

   a. Level - 0.5 volt peak ±3 db

   b. Input impedance - 75 ohms nominal

   c. Frequency - 41.25 to 47.25 MHz, band limited.

C. Control Inputs

1. Primary power (J6-2) (common J6-1)

   a. "ON" state - plus 5 ± 0.5 vdc into 1000 ohm load.

   b. "OFF" state - 0 ± 0.4 vdc.

   c. Rise time - less than or equal to 1 msec.

2. Subcarrier modulator (J6-3) (common J6-4)

   a. "ON" state - plus 5 ± 0.5 vdc into 1000 ohm load.

   b. "OFF" state - 0 ± 0.4 vdc.

   c. Rise time - less than or equal to 1 msec.

D. Monitor outputs

1. Primary exciter power ON/OFF (J6-6) (J6-5 common)

   a. "ON" state - plus 5 ± 0.5 vdc into 1000 ohm load.

   b. "OFF" state - 0 ±0.4 vdc

2. Subcarrier power ON/OFF (J6-7) (J6-8 common)

   a. "ON" state - plus 5 ± 0.5 vdc into 1000 ohm load.

   b. "OFF" state - 0 ±0.4 vdc

3. Modulation input present (J6-11) (J6-12 common)

   a. "ON" state - plus 5 ± 0.5 vdc into 1000 ohm load.

   b. "OFF" state - 0 ± 0.4 vdc.

4. Modulation output present (J6-13) (J6-14 common)

   a. "ON" state - plus 5 ± 0.5 vdc into 1000 ohm load.

   b. "OFF" state - 0 ± 0.4 vdc.

5. RF output power monitor (J6-9) (J6-10 common)
   a. Level varies from 0 to +5 vdc as the output power varies from 5 to 50 mw.
   b. Load impedance - TBD.

Circuit Description

Primary Power (TB1) — Application of primary power to the control and indicator unit is controlled by circuit breakers. When power is present on J8 and the circuit breakers are closed, DS2 will turn on indicating 115 vac power has been applied to the unit. Application of 115 vac power in this manner energizes the 115 vac 28 vdc converter, and power is available at the input terminal of TB1. The exciter may also be energized by the application of 28 vdc power to J7. Power from these two sources, J7 and J8, is combined into one source of power in TB1.

Subcarrier Modulator Switch (TB2) — The subcarrier modulator switch controls the application of -20 vdc power to the SCM.

Television Demodulator — The television demodulator receives a band limited input signal of 41.25 to 47.25 MHz at a level of 0.5 volt peak. The input signal (J1) is similar to that present in a television receiver intermediate frequency (IF) amplifier. The demodulator detects the video signal in the amplitude (AM) detector and the voice signal in the frequency modulation (FM) detector. For a more detailed discussion refer to the section on the TV demodulator.

Signal Combiners — The three signal combiners receive the five modulation inputs and condition and combine them to obtain the required output level and impedance to feed the modulation amplifier in the upper unit and subsequent deviation of the main carrier.

4.5-MHz Subcarrier Modulator — The subcarrier modulator is modulated by voice signal from an external source (J5) or from the TV voice output from the TV Demodulator. The frequency-modulated output is used to frequency deviate the exciter output carrier. The unit consists of an 11 MHz oscillator, which is frequency-modulated by the incoming voice signal. The 11 MHz signal is mixed with a 15.50 MHz crystal-controlled signal to produce an output carrier of 4.5 MHz. An AFC loop operates on the 11 MHz FM oscillator to maintain the frequency constant.

Modulation Presence Detector – The presence of modulation at the input to the power oscillator and at the output of the 2 GHz amplifier is detected by this subassembly.

The input modulation signal is provided by the test point on the output of the modulation amplifier. The output modulation signal is provided by the detected baseband output from the AFC unit. Except for the source of the signals, the operation of the two modulation detectors is the same.

5.1.4.8 Frequency Doubler No. 1 Subassembly

Purpose of Subassembly

Frequency doubler No. 1 is used in the microwave transmitter to double the frequency output of the 2 GHz amplifier. The unit converts a 5 watt input in the 1700 to 1940 MHz range to a 3 watt output in the 3520 to 3800 MHz frequency range.

Subassembly Description

The unit is a machined aluminum assembly with input and output SMA coaxial connectors and is mounted within the RF subsystem cabinet. Unit dimensions are 1 by 3-1/2 inches. The electrical characteristics are as follows:

Power requirements: none, passive unit.
Input and output impedance: 50 ohms
Input and output frequency range: Input: 1760 to 1940 MHz
Output: 3520 to 3880 MHz
Input power: 5 watts (+7 dbw) minimum
7 watts (+8.5 dbw) maximum
Output power: 3 watts (+4.8 dbw) minimum
Harmonics of input at output: -10 dB maximum (relative to desired X2 output)

Circuit Description

The unit consists of coaxial input and output matching and filter networks, two dc blocking capacitors, varactor diode, and selective resistor. The unit requires a 5 watt input signal to produce a 3 watt output signal at twice the input frequency.

5.1.4.9 Frequency Doubler No. 2 Subassembly

Purpose of Subassembly

The frequency doubler converts a 3 watt input in the 3527 to 3875 MHz range to a 1 watt output in the 7055 to 7750 MHz range.

Subassembly Description

This unit is a machined aluminum assembly with input and output OSM coaxial connectors and is mounted within the RF subsystem cabinet. Unit dimensions are 1 by 1 by 3-1/2 inches. The electrical characteristics are as follows:

Power requirements:		None, passive unit
Input and output impedance:		50 ohms
Frequency range:	input:	3527 to 3875 MHz
	output:	7055 to 7750 MHz

Input power: 2.5 watts (+4 dbw) minimum, 4 watts (+6 dbw) maximum

Output power: 1 watt (0 dbw) minimum

Harmonics of input at output: -10 db maximum (relative to desired X2 output)

Circuit Description

This unit consists of coaxial input and output matching and filter networks, two dc blocking capacitors, varactor diode, feedthrough capacitor, and two resistors. The unit requires a 2.5-watt input signal to produce a 1 watt output signal at twice the input frequency.

5. 1. 4. 10 Frequency Doubler No. 3 Subassembly

Purpose of Subassembly

This frequency doubler is used in the microwave transmitter to provide the final output signal. The unit converts a 1 watt input at 7. 250 GHz ± 100 MHz to a 250 milliwatt output at 14, 500 GHz ± 200MHz.

Subassembly Description

The unit is a wave-guide assembly with input and output waveguide connectors. The assembly is mounted in the RF subsystem cabinet. The unit's dimensions are 2. 12 by 5. 19 by 5. 38 inches. The electrical characteristics are as follows:

Power requirements: -20 vdc at 50 ma

Input output impedance: 50 ohms

Input andoutput frequency range: input: 7. 250 GHz
output: 14. 500 GHz

Input power: 1 watt (0 dbw) minimum

Output power: 0. 250 watt (-6 dbw) minimum

Harmonics of input at the output: All spurious and harmonically related outputs except the desired are 60 db below the output level, after the output filter.

Circuit Description

The unit consists of waveguide input and output matching and load isolators, varactor diode, and an output filter. The doubler requires a 1 watt input to produce a 0. 250 watt output.

5. 1. 4. 11 Subsystem Control Assembly

Purpose of Subassembly

The subsystem control unit provides meter test facilities for monitoring relative microwave transmitter performance.

Subassembly Description

The unit consists of a transistorized, etched-circuit monitor/ relay card contained in a chromate-coated aluminum box that is mounted on the RF subsystem cabinet door. A meter and test/selector switches are mounted on the module front panel. Unit dimensions are 4 by 5 by 3 inches. The electrical characteristics are as follows:

Power requirements:	-20 ±1 volt at 145 ma, maximum
Alarm outputs:	Phase-lock alarm Fault alarm (low RF power) Subsystem alarm Low transmitter RF power alarm

Circuit Description

The subsystem control unit provides test and monitoring facilities for a transceiver subsystem and furnishes switching logic and alarm grounds for selected functions. The unit monitors deviation level, transmit frequency and transmit power. The unit contains a plug-in monitor/relay card and a meter.

The front panel of the unit contains performance meter M1 and three switches. Selector switch S1 selects the transmit or receive function to be monitored by meter M1. AFC switch S2 (normally set to ON) is used to disable the AFC action during test and alignment. When the AFC switch is set to OFF, the subsystem alarm lamp lights. The 70 MHz SEL switch (S3) is used to select a 70 MHz signal from either the transmitter or receiver for comparative deviation testing. With the 70 MHz SEL switch set to RCV, the AFC is disabled, but the subsystem alarm lamp does not light if the AFC switch is in the ON position.

The DEV TEST position of selector switch S1 is used only if the system employs a pilot tone and is equipped with either a switching subsystem or a pilot detector. When S1 is in the DEV TEST, the 70 MHz SEL switch can be used to select a 70 MHz signal from the transmitter for comparative deviation testing.

If switch S1 is placed in the XMT FREQ position, a dc voltage from the AFC unit is applied to the meter. This dc voltage provides a meter indication of transmit frequency. The DSCRM ZERO control on the AFC unit is adjusted for a zero indication on the control unit meter when the transmitter is exactly on frequency. Any subsequent off-frequency condition is indicated by a meter indication above or below the zero reference.

5.1.4.12 Television Demodulator Subassembly

Purpose of Subassembly

The television demodulator is used in the K_u-Band exciter to demodulate the incoming television intermediate frequency (TV IF) into its video and audio components. The detected audio is used to frequency deviate a 4.5 MHz subcarrier oscillator. The 4.5 MHz SCO is recombined with the baseband video signal, and the combined signal is used to frequency modulate the main carrier.

Subassembly Description

The unit is a transistorized, metal circuit card contained in an aluminum box, which mounts in slides in the control and indicator unit. Unit dimensions are 7 by 6-1/16 by 1-1/4 inches. The electrical characteristics are as follows:

Power requirements:		24 vdc at TBD ma	
Input level:		0.5 volt peak ± TBD volt	
Input frequency:		41.25 to 47.25 MHz, similar to a television intermediate frequency IF signal. The input signal should be band limited to eliminate any adjacent channel interference.	
Input impedance:		75 ohms nominal	
Output impedance:		video:	75 ohms nominal
		audio:	600 ohms balanced
Output level:	video:	TBD	
	audio:	TBD	
Output frequency:		video:	100 Hz to 4.5 MHz ± TBD db
		audio:	TBD
Distortion:	video:	TBD	
	audio:	TBD	

Circuit Description

The Television (TV) Demodulator consists of an RF amplifier, an AM (video) detector, a video amplifier section, an FM detector, and an audio amplifier section.

The input signal is a TV IF in a frequency band of 41.25 to 47.25 MHz. The entire spectrum is amplified and passes to the AM detector through a double tuned circuit.

The incoming spectrum will appear as shown in Figure 5-3. The sound subcarrier is recovered by mixing the picture carrier (45.75 MHz) and the sound carrier (41.25 MHz) in the AM detector to obtain a 4.5 MHz beat note. The 4.5 MHz sound is trapped out of the video by a parallel trap. A series resonate circuit is used to feed the 4.5 MHz subcarrier into the FM detector.

The output contains a deemphasis network. This network is constructed such that the received TV audio, which is preemphasized, will give a flat response when detected. The audio output is then fed into the 4.5 MHz subcarrier modulator.

The audio output stages after the FM Detector are controlled by a switch. This switch provides for the interruption of dc power to the audio output to permit the use of the 4.5 MHz SCM with other audio signals while the video is being transmitted.

5.1.4.13 Subcarrier Modulator Assembly

Purpose of Subassembly

The subcarrier modulator (SCM) is used to frequency-deviate the K_u-Band exciter output. The output of the 4.5 MHz subcarrier modulator is frequency deviated with a voice signal from either the demodulated television or the external voice input. The SCM also has the capability to put preemphasis on the voice signal.

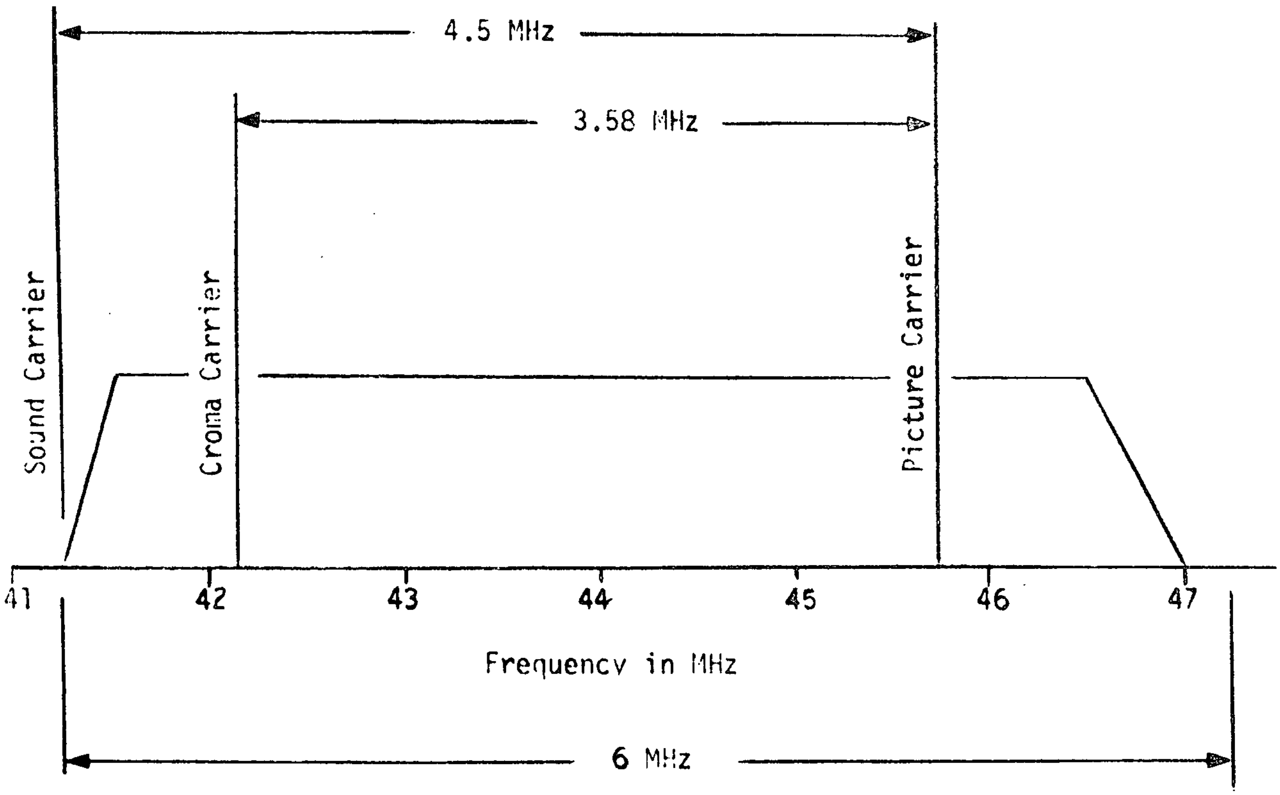


Figure 5-3. Input Spectrum for TV Demodulator

Subassembly Description

This unit is contained in an enclosed module 9-1/4 by 6-1/16 by 1-1/4 inches. Components are mounted on a two-layer etched circuit card, which is 5-3/4 by 7-4/5 inches. The unit is constructed for mounting in a 7-inch rack-mounted card cage. The electrical characteristics are as follows:

Power requirements: -20 to -22 vdc at 100 ma.

Input impedance: 600 ohms balanced.

Input level: -7 to + 12 dbm

Preemphasis: 75 μsec or flat (no emphasis), selectable by internal switching.

Modulation frequency response: audio channel response within 3 db from 50 Hz to 15 KHz.
Wide channel response (no emphasis) within 3 db from 50 Hz to 100 KHz.

Distortion: audio channel - NMT 1.5 percent
wide channel - NMT 3 percent
Distortion for ± 150 kHz; carrier deviation shall not exceed 1.5 percent for a 1 KHz test tone.

Output impedance: 75 ohms unbalanced or 1 kilohm, selectable by means of internal strapping option.

Carrier output level: adjustable from 20 to 300 mv P-P.

Deviation sensitivity: ±75 KHz linear deviation with a 1 KHz, -7 dbm test tone input.
Capable of ±150 KHz deviation with a 0 dbm test tone input.

Carrier output frequency: 4.5 MHz ±2 KHz at room ambient.

Circuit Description

The subcarrier modulator consists of a 600 ohm balanced input amplifier, which frequency deviates an 11 MHz LC oscillator. The oscillator frequency is controlled by an AFC loop. The output of the oscillator is fed to a mixer where it is mixed with a 15.50 MHz signal from a crystal controlled oscillator. The 4.5 MHz mixer output is fed through a filter, amplified, and fed to the output connector.

5.2 K_u-BAND POWER AMPLIFIER

A description of the K_u-Band power amplifier to be delivered under the Information Management System (IMS) Special Emphasis task is provided in this section.

5.2.1 Purpose of Unit

The K_u-Band power amplifier provides for amplification of the modulated signal generated by the K_u-band exciter described in Section 5.1. The power amplifier provides a power output of not less than 10 watts at a frequency of 14.5 GHz. The unit is capable of being mounted in a standard 19-inch equipment rack.

5.2.2 Description of Unit

The K_u-Band power amplifier selected is an off-the-shelf traveling-wave tube (TWT) amplifier, model number 1177HO4R-000, manufactured by the Electron Dynamics Division of Hughes Aircraft Company. A pictorial representation of the unit is shown in Figure 5-4. The electrical, mechanical, and environment specifications are tabulated in Table 5-2. Typical saturated power output and saturated gain versus frequency plots are presented in Figure 5-5. The power output curve shows that an RF power output of approximately 15 watts can be expected at 14.5 GHz.

5.2.3 Functional Characteristics and Performance

5.2.3.1 Input Versus Output Characteristics

A plot of the RF output power versus the RF input drive at a given frequency is called a "gain transfer" or "compression" curve. A sample is shown in Figure 5-6.

There are two regions where the TWT can be operated properly. They are the linear and saturation areas, as indicated in Figure 5-6. The linear region is that part of the compression curve where the gain of the amplifier will be the highest and is called "small signal gain." As the amplifier is driven toward saturation, the gain will be compressed to the point where further increases in the RF input drive will not result in a corresponding

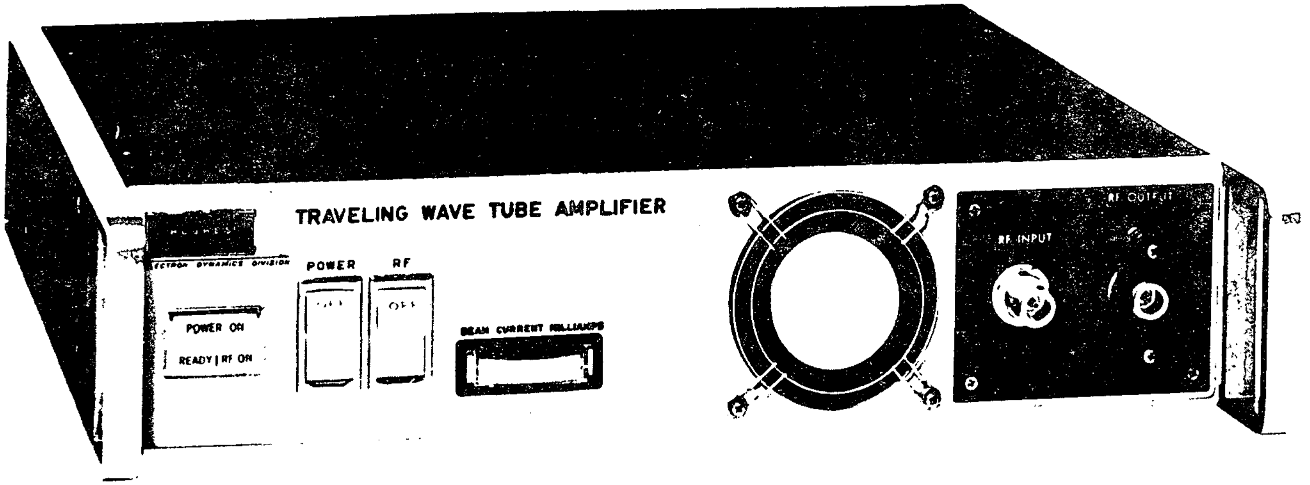
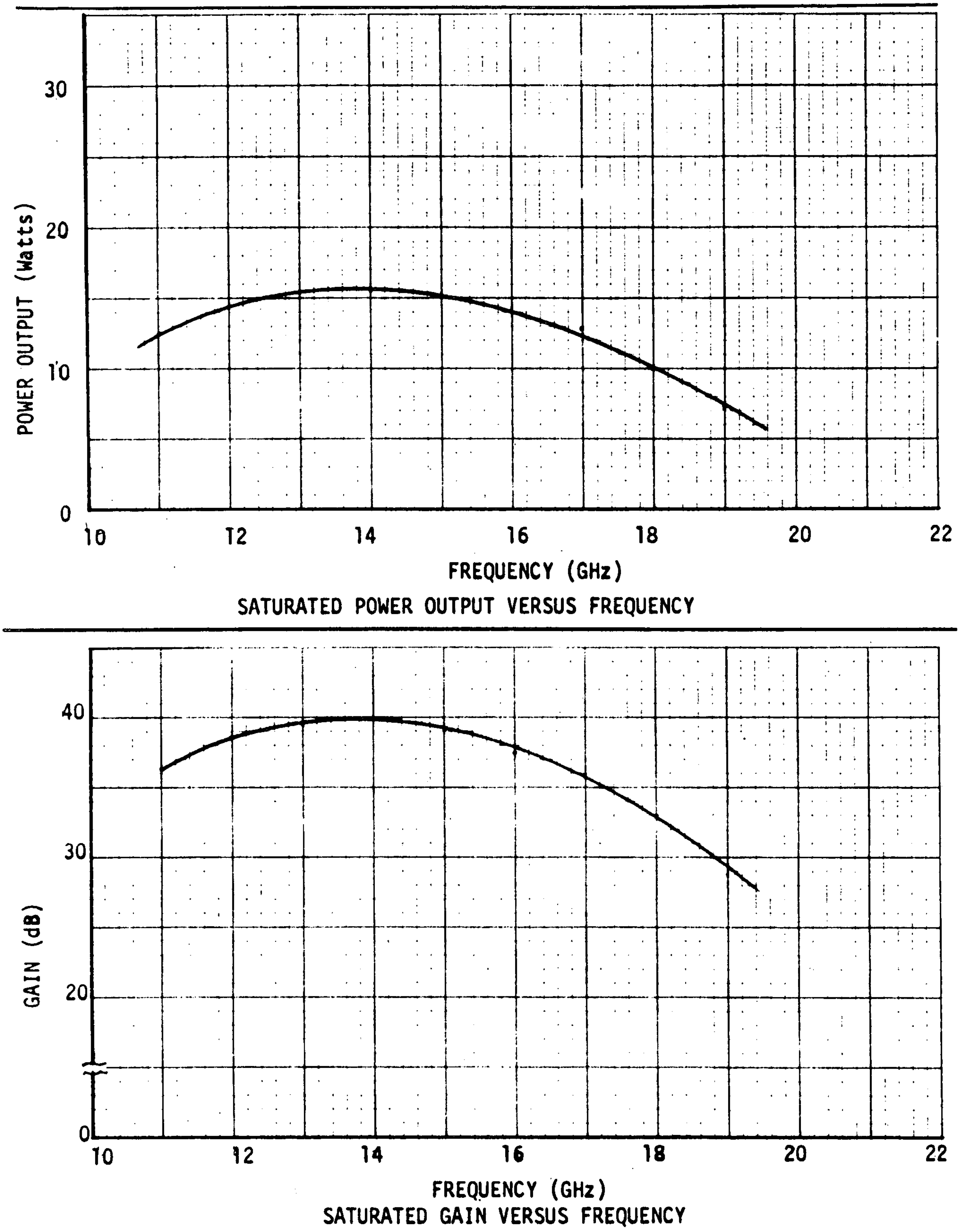


Figure 5-4. K_u-Band Power Amplifier

Table 5-2

SPECIFICATIONS

<u>Electrical</u>	
Power output	10 watts (minimum)
Output frequency	12.4 to 18.0 GHz
Gain at rated power output	30 dB minimum
Duty	CW
Input voltage	120 vac ±10 percent
Input frequency	50/60 Hz
Power consumption	250 watts
Noise figure	35 db maximum
Spurious modulation	-35 db minimum
VSWR	3.0:1
RF impedance	50 ohms
<u>Mechanical</u>	
Length	15.5 inches
Width	16.75 inches
Height	3.5 inches
Weight	20 lb maximum
Connector	WR 62 waveguide with UG 419/U waveguide flange.
<u>Environmental</u>	
Operating temperature	0 - 50° C ambient



SATURATED POWER OUTPUT VERSUS FREQUENCY

SATURATED GAIN VERSUS FREQUENCY

Figure 5-5. Typical Frequency Plots

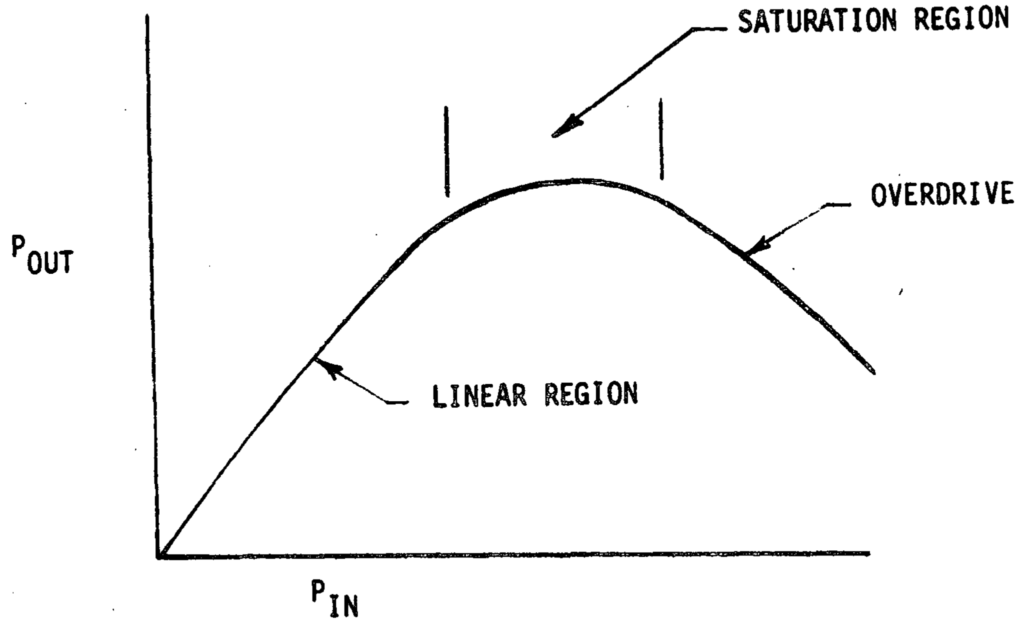


Figure 5-6. Gain Transfer Curve

increase in the RF output. In this area, the amplifier will provide its maximum RF output at the particular frequency and is operating at "large signal" or saturation*. It should be noted that the saturation area is rather broad because of the gain compression. If the RF drive is further increased, the amplifier's RF power output will begin decreasing and will be operating in the "overdrive" area. In this region, the TWT's beam will begin defocusing with increased current intercepting the helix structure. The TWT may be damaged by excessive overdrive for any length of time.

5.2.3.2 Noise Performance

Noise in the amplifier consists of three main components: KTB, AM, and FM or PM noise. KTB noise is generated by the heat generation into a 50 ohm equivalent impedance of the TWT and is -114 dbm/MHz. This

*Large signal gain is typically 6 db less than small signal gain because of the gain compression.

noise level will be increased by the noise figure and the small signal gain of the amplifier. Typically, these numbers are 30 db and 40 db, respectively; hence in normal operation, the thermal noise out of the amplifier will be typically -44 dbm/MHz when the amplifier is not saturated or without RF signal present.

The PM or FM noise is generated by ripple voltage on the cathode or beam power supply and consists of noise lines at the power line and converter switching frequencies, or their multiples. This noise level is typically 40 db below the carrier at saturation. Additional power supply filtering can be supplied (as an option) reducing this level to greater than 60 db below a carrier at saturation.

The AM noise is typically 20 db below the FM noise or 60 db below a saturated carrier. If the AM noise is critical, the amplifier should be operated at saturation to minimize AM components from the RF source. Figure 5-6 shows that small-level changes on the RF drive will not essentially effect the RF output when saturated.

Section 6
ELECTRONIC PACKAGING AND INSTALLATION

The 10-year lifetime requirement on Modular Space Station places a heavy premium on equipment operational longevity. To achieve this objective onboard equipment checkout and fault isolation and efficient replacement of failed items are required. Enhancement of on-orbit maintainability and a reduction in subsystem hardware costs can be achieved through use of a standardized electronic packaging and installation approach. This section summarizes a trade study on electronic equipment packaging and installation concepts and describes an approach that allows for fast replacement of failed items and cost reductions through use of a family of standardized modules. The selected approach is applicable for high-density electronic installations located throughout the ISS.

6.1 REQUIREMENTS

Electronic equipments are located in all modules of the ISS and fall into two basic installation categories:

A. Equipment mounted into consoles and racks located next to the vehicle exterior surfaces.

B. Equipment mounted into consoles and racks located adjacent to walkways or interior walls.

An example of category 1 installations is in the GPL, where many of the experiment operating consoles are located against the exterior wall. Examples of the second category are the electronic equipment enclosures in the center area of the GPL, and the primary command and control center in the crew/operations module.

It is significant to note that the category 1 installations do not afford direct rear access to the equipment enclosures. The book shelf packaging and installation concept developed in the 33-ft Space Station Study is therefore not directly applicable for Modular Space Station, as this approach was predicated upon both front and back accessibility to the equipment enclosures, (equipment access from the front and wiring access from the back). A design concept whereby access to electronic replaceable modules as well as to console wiring from a single side (plane) satisfies both installation categories and also allows two single-plane installation sections to be back-to-back, thus saving space.

A modular installation section (MIS) was selected as the basic installation unit for the study. A modular installation section is defined as a structure support system capable of accommodating individual (or groups of) Line replaceable units (LRU's) (modules), an interconnecting wiring system, and a heat transfer system consistent with minimum-weight and volume-usage goals. The LRU's consist of modules (replaceable assemblies) in a standardized family of sizes, which are inserted into the modular installation section.

6.1.1 Modular Packaging Requirements

In arriving at a standardized electronic modular packaging concept, the following requirements and objectives were considered:

A. The modular installation section shall have a structural, wiring, and heat-transfer system interface with the vehicle such that the section can be moved while in orbit to allow access for visual inspection and repair of the interior surface of the outer vehicle wall without impairing critical systems operations.

B. Provision shall be made for direct visual and physical access by the crew for replacement, removal, or servicing of the replaceable assemblies and for disconnecting or connecting related electrical connectors.

C. Access to the wire harnesses for modifications or repair shall be provided. This access is to be from the same plane from which removal of the replaceable assembly is achieved.

D. Keying techniques shall be provided to preclude inadvertent mislocation and improper connector mating of the replaceable assembly.

E. The design shall be compatible with the crew limitations relative to maximum torquing forces, connector extraction forces, adjustment access requirements, assembly form factors, guides, and other constraints.

F. Refurbishable or replaceable captive fasteners shall be used for each piece of equipment that has a planned maintenance capability.

G. The order of precedence for in-orbit maintenance tasks shall be (1) shirtsleeve environment, (2) space suit (IVA), and (3) space suit (EVA). EVA tasks shall be of an emergency nature only.

H. The design shall be standardized to the extent that it can accommodate various replaceable assembly groups and so that the modular installation section can be efficiently used in various vehicle areas. Cost effectiveness shall be a consideration in the design selection. Commonality is a primary consideration. As a goal, common module installation structures, standardized mounting means, and the like shallbe considered.

I. The packaging design of the replaceable assembly shall be capable of accommodating multitechnology techniques; i. e., printed circuits, thick film, hybrid designs, and discrete components. The equipment shall be designed for maximum flexibility that will insure ability to function with new subsystems and technologies.

J. The MIS structure shall be designed to accommodate the following interface requirements:

   1. Provisions for mounting electrical and coolant line connectors compatible with the vehicle interface.
   2. Mounting attachments to the vehicle structural interface which enable the modular installation section assembly to be moved while in orbit to allow visual inspection of the vehicle's outer wall.
   3. Provisions at the replaceable assembly interface for mounting, keying, and interconnecting wiring.
   4. Installation of electrical interconnecting wire harnesses, including physical separation of EMI wiring categories. (Seven categories are assumed.)

5. Installation of a heat transfer system (i.e., cold plates, heat pipes, etc.) where applicable.
6. Provisions for partitioning the structural complex to inhibit flame propagation.

K. The section shall have provisions for accommodating the following safety requirements:

1. Electrical bonding (grounding) of the replaceable assembly, and support structure to vehicle ground.
2. A maximum touch temperature of 105° F for crew-exposed surfaces.
3. No exposed sharp edges, corners, or protrusions.
4. Flame propagation protection.
5. Compatibility with in-orbit maintenance tools for replaceable assembly removal and ease of handling.

   Each subsystem shall be designed so that major failures are repairable and to provide maximum ease of maintenance under expected astronaut skill. Precision elements will be provided with suitable guides and locking devices as aids in replacement. Refurbishable or replaceable captive fasteners shall be used for all equipment that has planned maintenance capability.
6. Redundant equipment physically separated, where possible, to minimize the probability of damage to one when the other is damaged.

6.2 TRADEOFF STUDY AND RATIONALE

The selection of a packaging and installation design concept was approached by constructing cardboard mockups of several candidate approaches. These mockups were evaluated in terms of several design parameters. Some of the design parameters were quantifiable while others were given subjective ratings. The trade study plan used in the evaluation is depicted in Figure 6-1.

6.2.1 Basic Replaceable Assembly

The basic replaceable assembly configuration was established in the initial study effort. This preceded installation section configuration studies. The

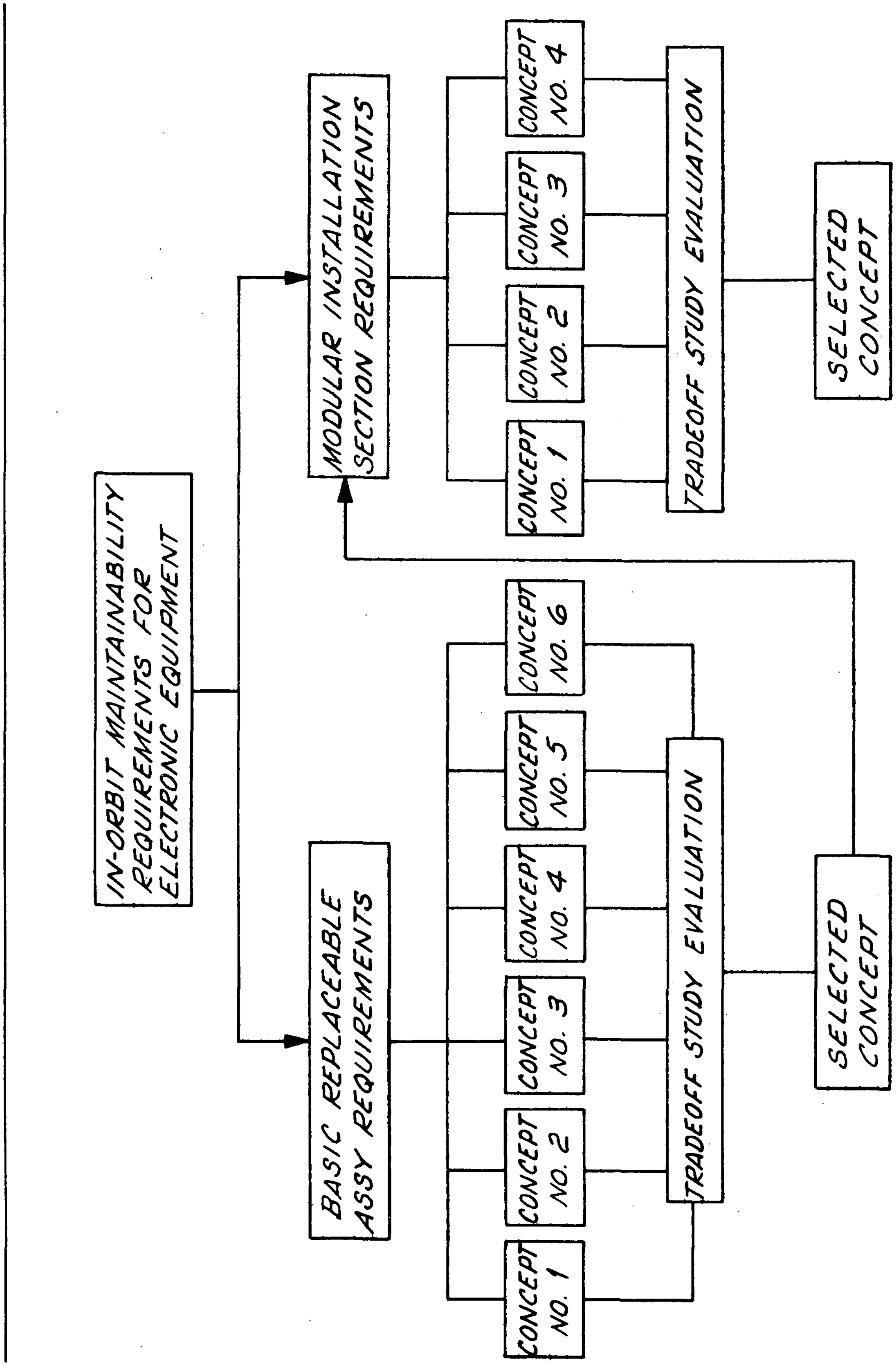

IN-ORBIT MAINTAINABILITY REQUIREMENTS FOR ELECTRONIC EQUIPMENT
BASIC REPLACEABLE ASSY REQUIREMENTS
CONCEPT NO. 1
CONCEPT NO. 2
CONCEPT NO. 3
CONCEPT NO. 4
CONCEPT NO. 5
CONCEPT NO. 6
TRADEOFF STUDY EVALUATION
SELECTED CONCEPT
MODULAR INSTALLATION SECTION REQUIREMENTS
CONCEPT NO. 1
CONCEPT NO. 2
CONCEPT NO. 3
CONCEPT NO. 4
TRADEOFF STUDY EVALUATION
SELECTED CONCEPT

Figure 6-1. Tradeoff Study Flow Chart

design factors and their selected solution and trades are given in subsequent paragraphs.

6.2.1.1 Form Factor

The assembly form factor is based on housing two 5.00- by 7.00-in. printed circuit boards, capable of accommodating about 150 discrete components and 70 to 100 wire terminations per board. The board area, component density and wire density is based on a survey of OWS and GSE printed circuit boards. Thick or thin film substrates could easily be accommodated on the printed circuit boards resulting in considerably higher component densities. Two printed circuit boards were chosen rather than one board in the basic LRU because the ratio of structural housing cost and weight per board is significantly less for two boards than for one board and because interface wiring would be minimized.

Based on this rationale, a basic LRU form factor, excluding connectors, of 9.00 in. long, 7.00 in. high, and 2.00 in. wide was selected. The assembly form factor is shown in Figure 6-2.

6.2.1.2 Connector Interface Capability

Based on a survey of OWS electronic assemblies it was determined that the basic LRU would have to accommodate four EMI wiring categories comprised of about 140 wires total. Therefore, a group of four connectors with a capability of terminating 140 wires is required. Concept layouts were made to confirm that the assembly form factor could accept such a connector group.

6.2.1.3 Connector Location Evaluation

Using the established form factor and connector requirements, six basic assembly concepts were evaluated relative to interconnection technique and connector location. The six concepts are shown in Figure 6-3 and are described as follows:

A. Concept 1—Connectors mounted on frontal or top surface of the assembly.

B. Concept 2—Connectors mounted on one side surface.

C. Concept 3—Connectors mounted on two side surfaces.

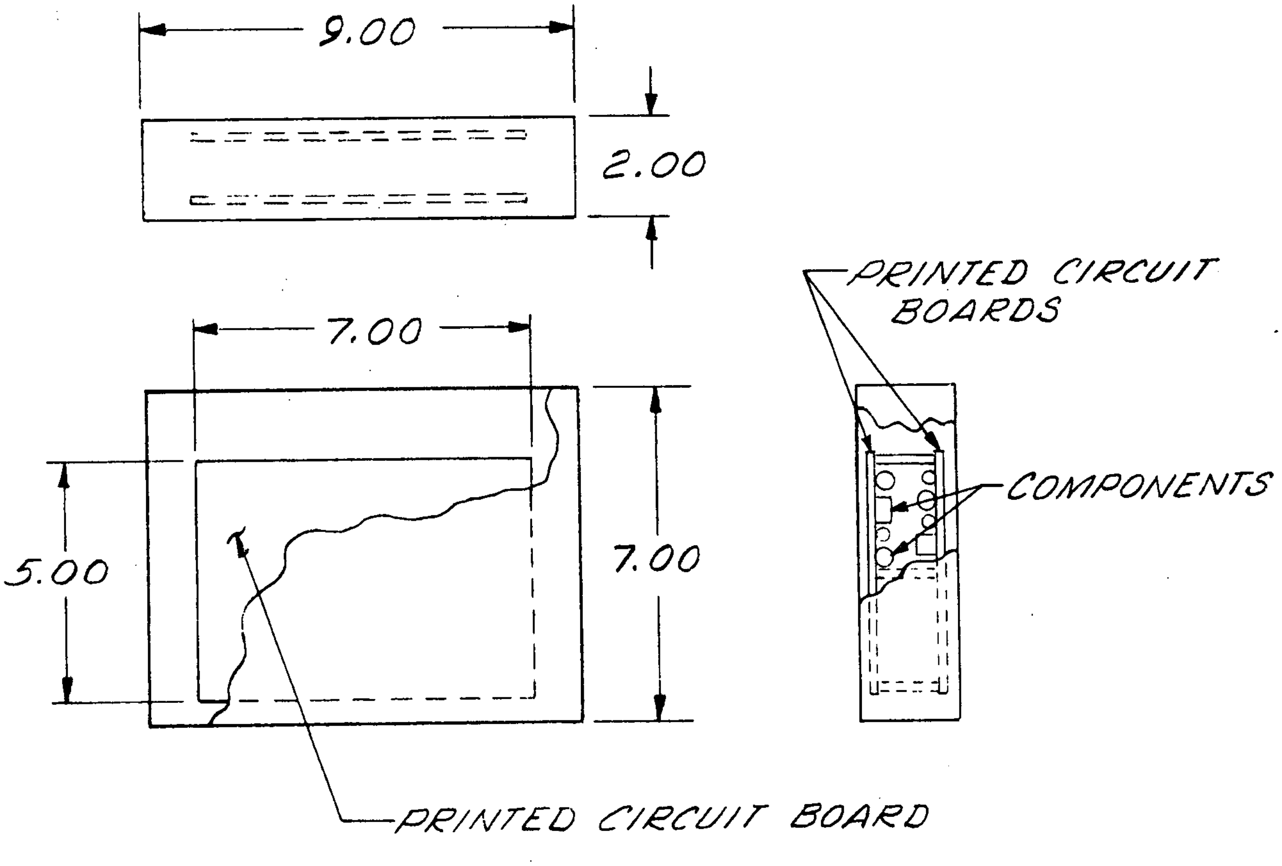


Figure 6-2. Basic Replaceable Assembly Form Factor

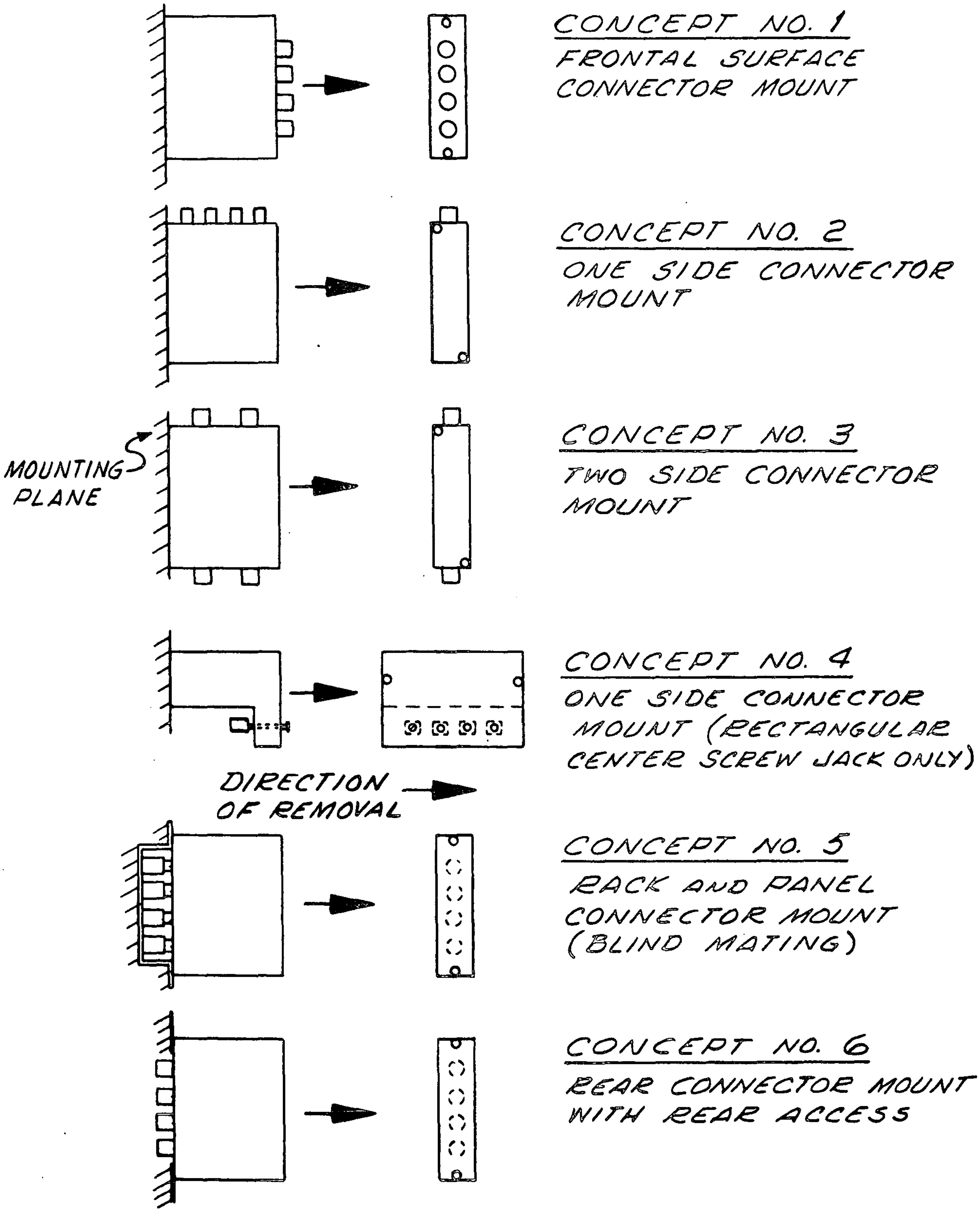

CONCEPT NO. 1
FRONTAL SURFACE CONNECTOR MOUNT
CONCEPT NO. 2
ONE SIDE CONNECTOR MOUNT
MOUNTING PLANE
CONCEPT NO. 3
TWO SIDE CONNECTOR MOUNT
CONCEPT NO. 4
ONE SIDE CONNECTOR MOUNT (RECTANGULAR CENTER SCREW JACK ONLY)
DIRECTION OF REMOVAL
CONCEPT NO. 5
RACK AND PANEL CONNECTOR MOUNT (BLIND MATING)
CONCEPT NO. 6
REAR CONNECTOR MOUNT WITH REAR ACCESS

Figure 6-3. Basic Replaceable Assembly Concepts

D. Concept 4—Connectors mounted on one long side of the assembly surface, using rectangular center screwjack connectors.

E. Concept 5—Connectors mounted on bottom surface of the assembly, with blind mating (no rear access).

F. Concept 6—Connectors mounted on bottom surface, with rear access for hand disengagement.

These concepts were evaluated, using both rectangular center screwjack connectors and round connectors. The rear mount connector methods, Concepts 4, 5, and 6, were discarded since the requirement for frontal wiring access and also for visual and physical access could not be met. The use of center screwjack connectors was eliminated primarily because of the additional development and qualification required for acceptance, even though they offer certain advantages over round connectors. Also, the round connectors represented the worst-case footprint, and therefore, rectangular connectors could easily be used in place of the round type if it became economically effective. Concept 3 was also discarded because it would require about a 30-percent greater installation footprint area than concept 2.

Concepts 1 and 2 were the remaining candidates, and they were evaluated in the installation section tradeoff study to evaluate the installation wiring system interface.

6.2.1.4 Mounting Techniques

Since replaceable captive mounting bolts with visual and physical access were defined as a requirement in the guidelines and constraints document, it was necessary to investigate several methods for applicability. A special bolt design with the following major features evolved:

A. Recessed hex socket head for easy wrenching.

B. Retraction spring for a positive and visual disengagement.

C. Two-piece 0.25-inch-diameter shaft with a coupler to enable it to be refurbished or replaced.

D. A special thread to minimize the number of turns to tighten.

Two mounting bolts per module were chosen as the minimum number required to provide secure mounting and good thermal characteristics.

6.2.1.5 Thermal Interface

A thermal analysis was performed on the replaceable assembly and installation assembly interfaces to determine allowable heat dissipation per module.

To determine the design adequacy of the thermal interface, certain conditions had to be defined.

1. Conditions
   - A. Coolant medium—water
   - B. Coolant flow rate: 500 lb/hr minimum and 960 lb/hr maximum
   - C. Coolant inlet temperature: 70° F
   - D. Coolant line size: 0.20-inch inside diameter
   - E. Assembly mounting:- two 10-32 bolts: 20 in-lb torque.
   - F. Assembly thermal contact area per bolt: 2.50 sq in.
   - G. Installation configuration: eight basic assemblies at 20 watts, each installed in a column on two "I" beam structural elements containing two coolant lines each.
   - H. All material is 6061 aluminum.

2. Analysis

   A thermal analysis was performed for the assembly closest to the coolant inlet source (Case 1) and also for the assembly farthest from the inlet source (Case 2) in the installation configuration. This was done for both minimum and maximum coolant flow rates. In each case, temperatures were calculated for the condition where the assembly had a uniform heating source and for the configuration having a single point heating source. The analysis was also performed, using Coolanol as the coolant medium, because this medium is used on Skylab equipment. The resultant temperatures are shown in the summary Table 6-1.

   Thermal resistances and Δ temperatures for the thermal interfaces were considered as follows:

   - A. Between fluid and coolant line wall.
   - B. Between coolant line wall and support structure.

Table 6-1

REPLACEABLE ASSEMBLY MAXIMUM TEMPERATURES

	Water Flow Rates		Coolanol Flow Rates	
	500 lb/hr	960 lb/hr	500 lb/hr	960 lb/hr
1. Assembly located at coolant inlet				
A. Assembly base temp (uniform heating)	155° F	153° F	184° F	181° F
B. Component case temp (point source heating)	251° F	249° F	280° F	277° F
2. Assembly located most remote from inlet				
A. Assembly base temp (uniform heating)	161° F	156° F	198° F	188° F
B. Component case temp (point source heating)	257° F	252° F	294° F	284° F

C. Between support structure and assembly base.

D. Between assembly base and a component located at a worst-case location for single point heating.

These interface Δ temperatures are shown in Figure 6-4.

3. Conclusions

The results of the thermal analysis indicate that 20-watt heat dissipation per replaceable assembly is reasonable. The water coolant system can adequately transfer this heat at a rate sufficient to allow internal components to operate within normal derated temperature valves.

6.2.2 Installation Section

Using the basic replaceable assembly as defined in previous paragraphs, several installation concepts were investigated. The most promising concepts were selected to be evaluated via tradeoff studies and cardboard mockups.

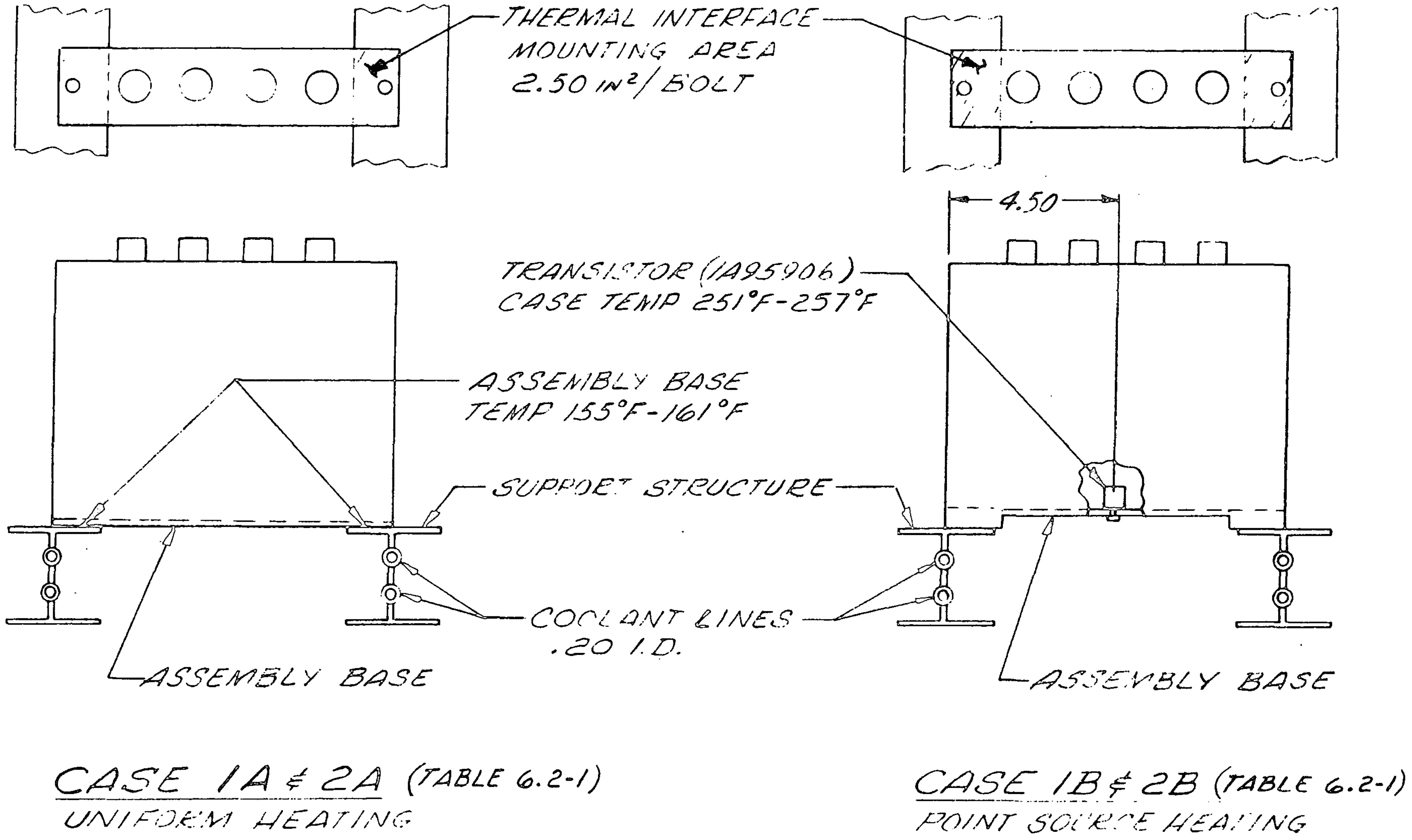


Figure 6-4. Replaceable Assembly Thermal Interfaces and Temperatures

6.2.2.1 Concept Descriptions

The installation section concept consists of an array of replaceable assemblies mounted on "I" beam structural supports that contain active coolant lines. The various wiring distribution and interconnecting schemes are shown in the respective concept drawings of Figures 6-5 through 6-8.

A brief description of each concept is as follows.

A. Concept 1

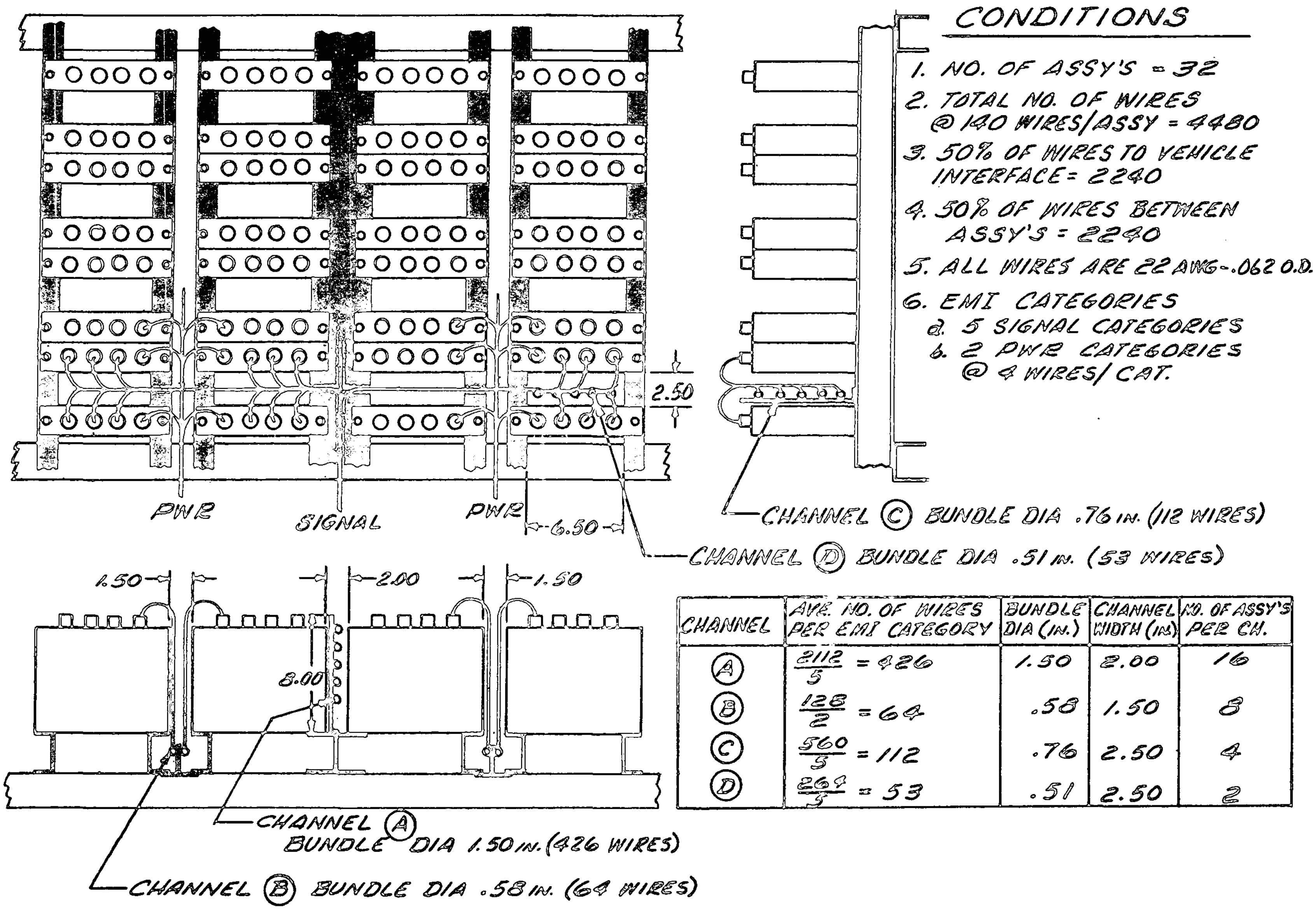
This configuration consists of four vertical rows of replaceable assemblies separated by three vertical wiring channels and four horizontal wiring channels, with wire bundles stacked vertically in the channels. The central vertical channel contains signal-type EMI wiring categories. This wiring is then routed into the horizontal channels and interconnected to a double horizontal row of replaceable assemblies, with front surface mounted connectors. The power wiring categories are routed in two separate vertical channels, which allow interconnections to be made to two double vertical rows of replaceable assemblies.

B. Concept 2

This concept consists of two vertical rows of replaceable assemblies with side-mounted connectors, separated by one central wiring channel containing all seven EMI wiring categories mounted in a flat plane. The interconnections are then made between the respective assembly and wire bundle in the central channel.

C. Concept 3

This arrangement is comprised of four vertical rows of replaceable assemblies separated vertically by a central wiring channel containing all seven EMI wiring categories. The assemblies are separated horizontally by horizontal wiring channels that contain all EMI categories to interconnect a double horizontal row of assemblies with front surface mounted connectors.

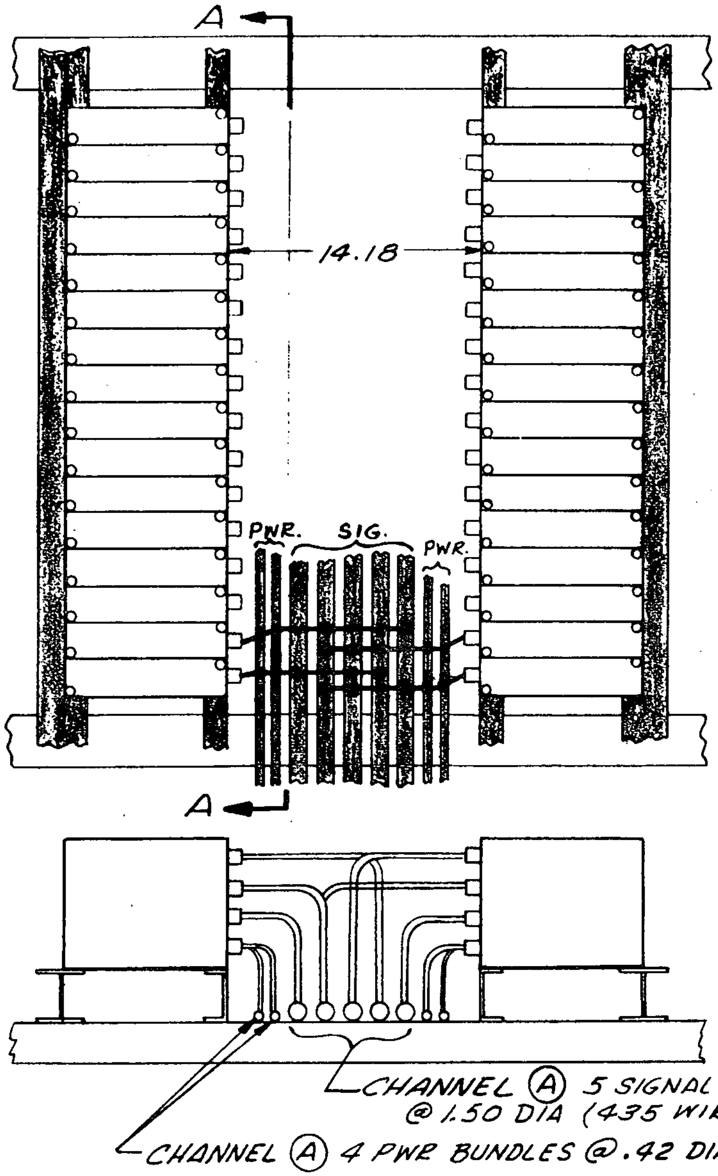

PWR
SIGNAL
PWR
6.50
2.50
1.50
2.00
1.50
8.00
CHANNEL Ⓒ BUNDLE DIA .76 IN. (112 WIRES)
CHANNEL Ⓓ BUNDLE DIA .51 IN. (53 WIRES)
CHANNEL Ⓐ BUNDLE DIA 1.50 IN. (426 WIRES)
CHANNEL Ⓑ BUNDLE DIA .58 IN. (64 WIRES)

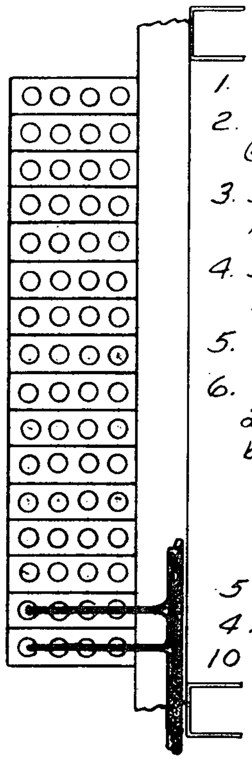
CONDITIONS

1. NO. OF ASSY'S = 32
2. TOTAL NO. OF WIRES @ 140 WIRES/ASSY = 4480
3. 50% OF WIRES TO VEHICLE INTERFACE = 2240
4. 50% OF WIRES BETWEEN ASSY'S = 2240
5. ALL WIRES ARE 22 AWG-.062 O.D.
6. EMI CATEGORIES
   a. 5 SIGNAL CATEGORIES
   b. 2 PWR CATEGORIES @ 4 WIRES/CAT.

CHANNEL	AVE. NO. OF WIRES PER EMI CATEGORY	BUNDLE DIA (IN.)	CHANNEL WIDTH (IN.)	NO. OF ASSY'S PER CH.
Ⓐ	$\frac{2112}{5} = 426$	1.50	2.00	16
Ⓑ	$\frac{128}{2} = 64$	.58	1.50	8
Ⓒ	$\frac{560}{5} = 112$	.76	2.50	4
Ⓓ	$\frac{264}{5} = 53$	.51	2.50	2

Figure 6-5. Concept No. 1


A
14.18
PWR.
SIG.
PWR.
A
CHANNEL (A) 5 SIGNAL BUNDLES @ 1.50 DIA (435 WIRES/BUNDLE)
CHANNEL (A) 4 PWR BUNDLES @ .42 DIA (32 WIRES/BUNDLE)



SECTION A-A

CONDITIONS

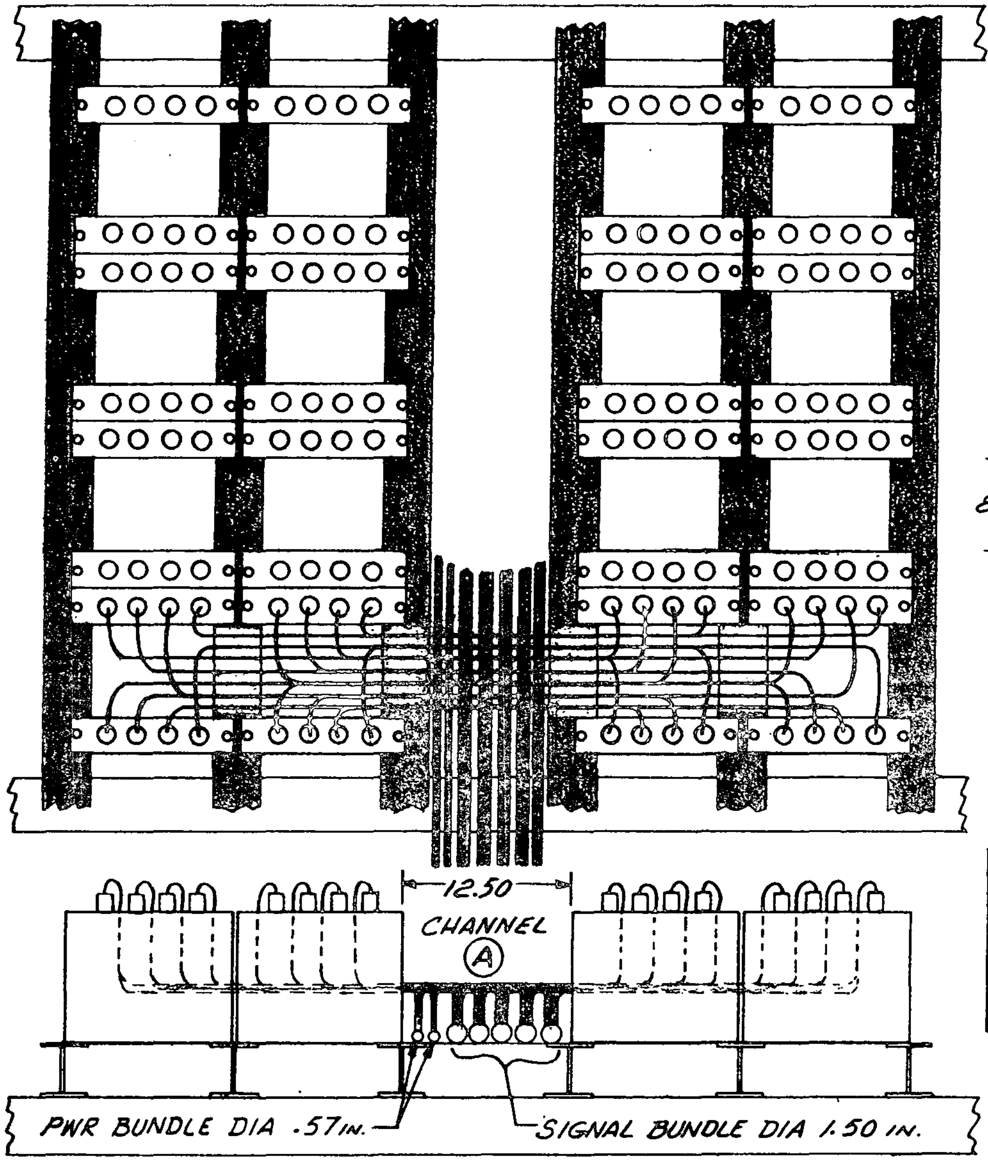
1. NO. OF ASSY'S = 32
2. TOTAL NO. OF WIRES @ 140 WIRES/ASSY = 4480
3. 50% OF WIRES TO VEHICLE INTERFACE = 2240
4. 50% OF WIRES BETWEEN ASSY'S = 2240
5. ALL WIRES 22 AWG. - .062 O.D.
6. EMI CATEGORIES
   a. 5 SIGNAL CATEGORIES
   b. 2 PWR CATEGORIES @ 4 WIRES PER CAT.

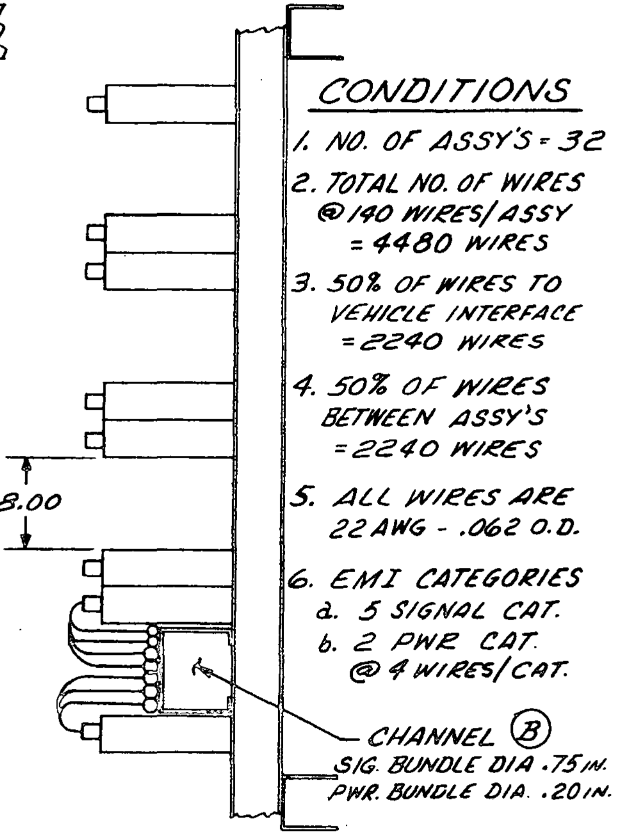
CHANNEL SIZING

5 SIG. BUNDLES @ 1.50 EA. = 7.50 IN.
4 PWR. BUNDLES @ .42 EA. = 1.68 IN.
10 BUNDLE SPACES @ .50 EA. = 5.00 IN.
CHANNEL WIDTH = 14.18 IN.

CHANNEL	AVE. NO. OF WIRES PER EMI CATEGORY	BUNDLE DIA. (IN.)	NO. OF ASSY'S PER CHANNEL
(A) SIG.	$\frac{2126}{5}$ = 435	1.50	32
(A) PWR.	$\frac{4(16)}{2}$ = 32	.42	

Figure 6-6. Concept No. 2



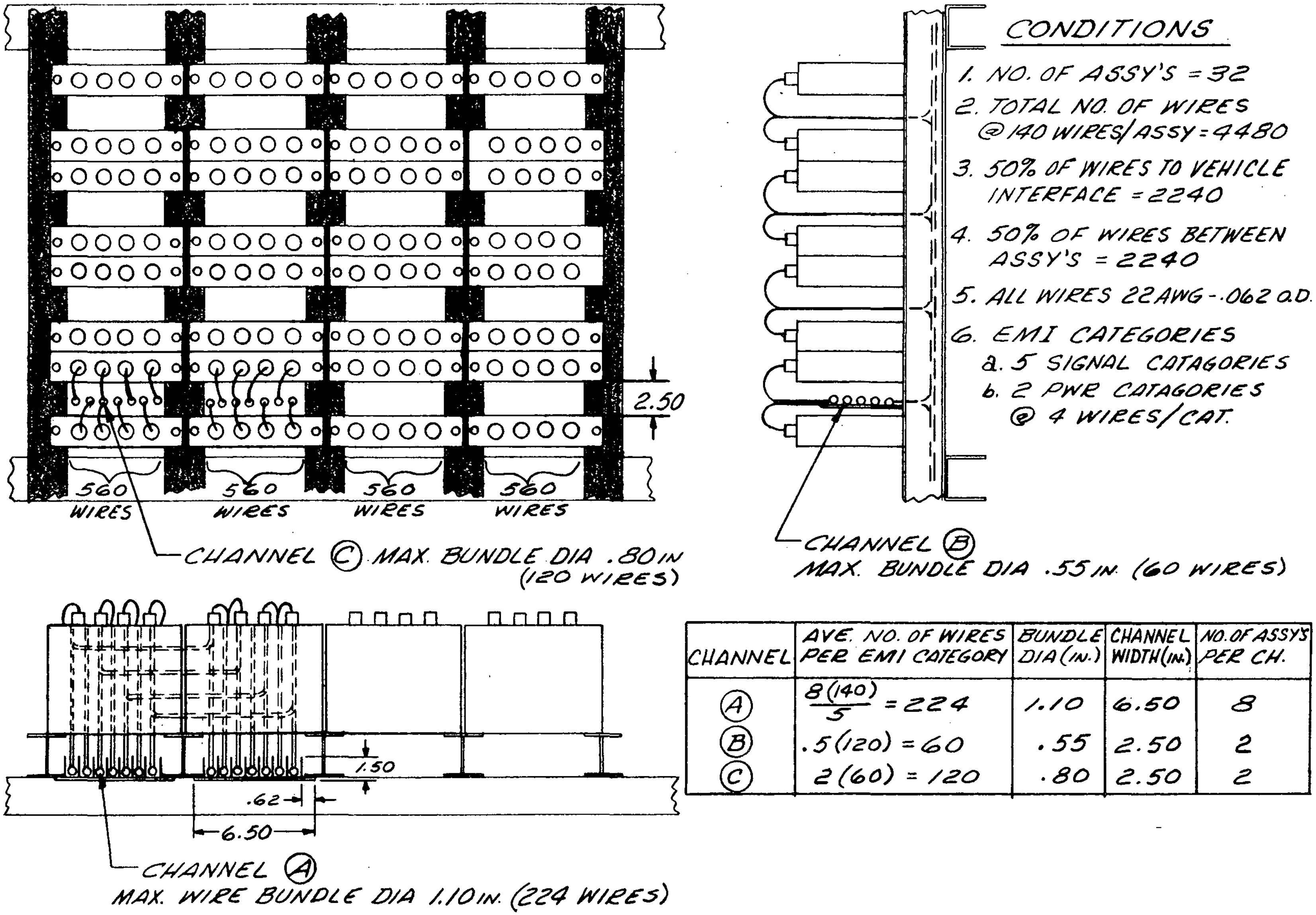


CONDITIONS

1. NO. OF ASSY'S = 32
2. TOTAL NO. OF WIRES @ 140 WIRES/ASSY = 4480 WIRES
3. 50% OF WIRES TO VEHICLE INTERFACE = 2240 WIRES
4. 50% OF WIRES BETWEEN ASSY'S = 2240 WIRES
5. ALL WIRES ARE 22 AWG - .062 O.D.
6. EMI CATEGORIES
   a. 5 SIGNAL CAT.
   b. 2 PWR CAT. @ 4 WIRES/CAT.

CHANNEL		AVE. NO. OF WIRES PER EMI CAT.	BUNDLE DIA (IN.)	NO. OF ASSY'S PER CH.
Ⓐ	SIG	435	1.50	8
	PWR	64	.57	
Ⓑ	SIG	109	.75	8
	PWR	16	.20	

Figure 6-7. Concept No. 3


560 WIRES
560 WIRES
560 WIRES
560 WIRES
2.50
CHANNEL Ⓒ MAX. BUNDLE DIA .80 IN (120 WIRES)
CHANNEL Ⓑ MAX. BUNDLE DIA .55 IN. (60 WIRES)
1.50
.62
6.50
CHANNEL Ⓐ MAX. WIRE BUNDLE DIA 1.10 IN. (224 WIRES)

CONDITIONS

1. NO. OF ASSY'S = 32
2. TOTAL NO. OF WIRES @ 140 WIRES/ASSY = 4480
3. 50% OF WIRES TO VEHICLE INTERFACE = 2240
4. 50% OF WIRES BETWEEN ASSY'S = 2240
5. ALL WIRES 22AWG - .062 O.D.
6. EMI CATEGORIES
   a. 5 SIGNAL CATAGORIES
   b. 2 PWR CATAGORIES @ 4 WIRES/CAT.

CHANNEL	AVE. NO. OF WIRES PER EMI CATEGORY	BUNDLE DIA (IN.)	CHANNEL WIDTH (IN.)	NO. OF ASSY'S PER CH.
Ⓐ	$\frac{8(140)}{5} = 224$	1.10	6.50	8
Ⓑ	.5(120) = 60	.55	2.50	2
Ⓒ	2(60) = 120	.80	2.50	2

Figure 6-8. Concept No. 4

D. Concept 4

This concept consists of four vertical rows of replaceable assemblies with connectors mounted on the front surface. Individual wiring channels containing all seven EMI wiring categories are located underneath each vertical row of assemblies. Each channel has metallic separators for shielding between EMI categories to enable closer bundle spacing to be achieved. The wires from the lower channel egress between every second assembly to provide the interconnection to two assemblies. The spacing between horizontal rows of assemblies would also be used for any cross wiring if required.

6.2.2.2 Tradeoff Study Assumptions

The following criteria was assumed as common to all of the installation concepts for the purpose of the tradeoff study.

A. The same replaceable assembly form factor excluding connectors is used in all installation assemblies.

B. Similar structure and cold-plate system.

C. A wiring system which will accommodate a complex of 32 basic replaceable assemblies with 70 of the 140 wires per assembly going to or from the installation section interface.

D. The same round connector family.

E. Flexing or bending of wires at the replaceable assembly interface when disconnecting or connecting the interface connectors during maintenance is considered acceptable from reliability standpoint. Extensive tests on previous programs substantiate this approach.

F. The use of a hand held tool to disconnect or connect the replaceable assembly connectors is acceptable if connector spacing does not allow for direct hand operations.

G. The possible removal of replaceable assemblies to gain access for installation assembly wiring modifications is acceptable since wiring maintenance is considered a secondary requirement.

H. The following tradeoff parameters are listed in order of highest weighting factor.

1. Ease of replaceable assembly replacement (crew time).
2. Weight
3. Volume usage.

I. Only seven EMI categories will be considered for the purpose of the tradeoff study.

6.2.2.3 Tradeoff Study Evaluation

The tradeoff study evaluation is summarized in Table 6-2. The important design parameters were compared and evaluated for each of the four installation section concepts. Based on these comparisons, the concepts were selected in the following order of desirability:

Concept 4
Concept 1
Concept 2
Concept 3

The prime advantages of Concept 4 over the other concepts are as follows:

A. Relative to Concept 1

1. Better utilization of volume and frontal area.
2. Easier to manage wire routing in the cross channels to mitigate potential EMI problems.
3. The capability of accommodating multiple-form factor increments of the basic replaceable assembly in both the horizontal and vertical directions is feasible whereas this is not feasible in the horizontal direction for Concept 1 without blocking off main wiring channels.
4. Ease of fabrication of wire harnesses since there are fewer wire bends per wire and each harness would be less complex in geometry.

B. Relative to Concept 2

1. Better utilization of frontal area.
2. Better access to the replaceable assembly connectors, resulting in less crew time and reduced risk of wire damage during assembly replacement. Concept 2, however, could have as many as eight connectors and the respective cables within each 2-inch increment of the main wiring channel resulting in excessively high wiring density, which prevents easy access.

Table 6-2
INSTALLATION STUDY EVALUATION SUMMARY

Design Parameters	Concept 1	Concept 2	Concept 3	Concept 4
1. Percent volume usage efficiency, assembly volume	28 percent	40 percent	19 percent	33 percent
Assembly and structural and wiring volume				
2. Percent frontal area usage efficiency assembly area	50 percent	56 percent	35 percent	62 percent
Assembly and wiring				
3. Connector access	Good	Very difficult	Good	Good
4. Wiring access	Difficult, may require assembly removal	Very difficult, may require assembly removal	Excellent, assembly removal not required	Difficult, may require assembly removal
5. Capability of EMI mitigation	Fair	Difficult	Good	Good
6. Wire support complexity, number of bracket types	3	1	2	2
7. Ease of harness fabrication	Fair	Very difficult	Good	Good
8. Structure				
a. Installation sizes	Good	Good	Good	Good
b. Assembly sizes	Fair	Fair	Fair	Excellent
9. Capability of cross channel wiring	Good	Excellent	Excellent	Good
10. Ease of assembly removal	Good	Difficult (connector access)	Good	Good
11. Weight (lb)				
Assembly	5. 00	5. 00	5. 00	5. 00
Related structural per assembly	0. 32	0. 17	0. 38	0. 27
Related wire supports per assembly	0. 64	0. 58	1. 60	0. 55
Total (lb)	5. 96	5. 75	6. 98	5. 82

3. Easier to manage wire routing to minimize potential EMI problems. Concept 2 would overlay many cables to the assembly, crossing over each other.
4. Ease of fabrication of wire harnesses because wiring density and complexity are much less.

C. Relative to Concept 3

1. Much better utilization of volume and frontal area.
2. Easier to manage wire routing in the cross channels to mitigate potential EMI problems. Concept 3 could have the assembly cables crossing each other.
3. Less support structure weight per replaceable assembly. Concept 3 has wider cross channels, resulting in a need for longer lengths of support structure.

The major disadvantage of Concept 4 is that the replaceable assemblies must be removed as required to effect any major wiring maintenance. However, since wiring maintenance is considered a secondary requirement, this parameter is not a major weighting factor.

6.3 SELECTED DESIGN CONCEPT

Based on the evaluation of the design tradeoff study, Concept 4 was selected as the optimum concept for meeting Modular Space Station requirements. The total installation section concept is shown in Figure 6-9.

A description of the major features of the selected concept is as follows:

6.3.1 Modular Installation Section

The modular installation section (see Figure 6-9) consists of a support structure capable of accommodating individual or groups of replaceable assemblies, an interconnecting wiring system, and a heat-transfer system with access from a common plane for in-orbit maintainability of the assemblies and wiring.

6.3.1.1 Installation Section Support Structure

The basic structural elements, Figure 6-10, consist of an "I" beam shaped extrusion with integral coolant lines for mounting and cooling of replaceable

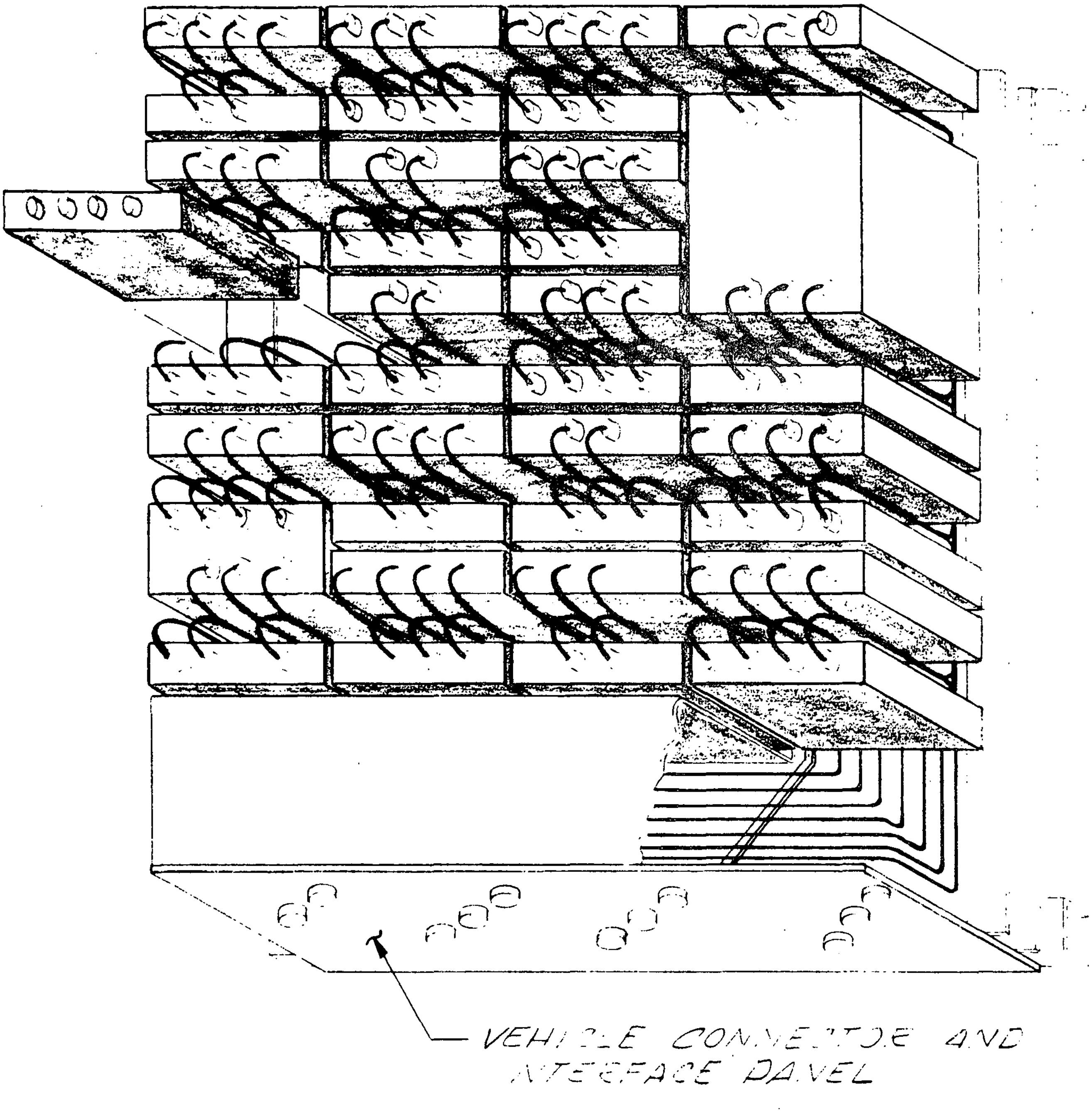


Figure 6-9. Modular Installation Section

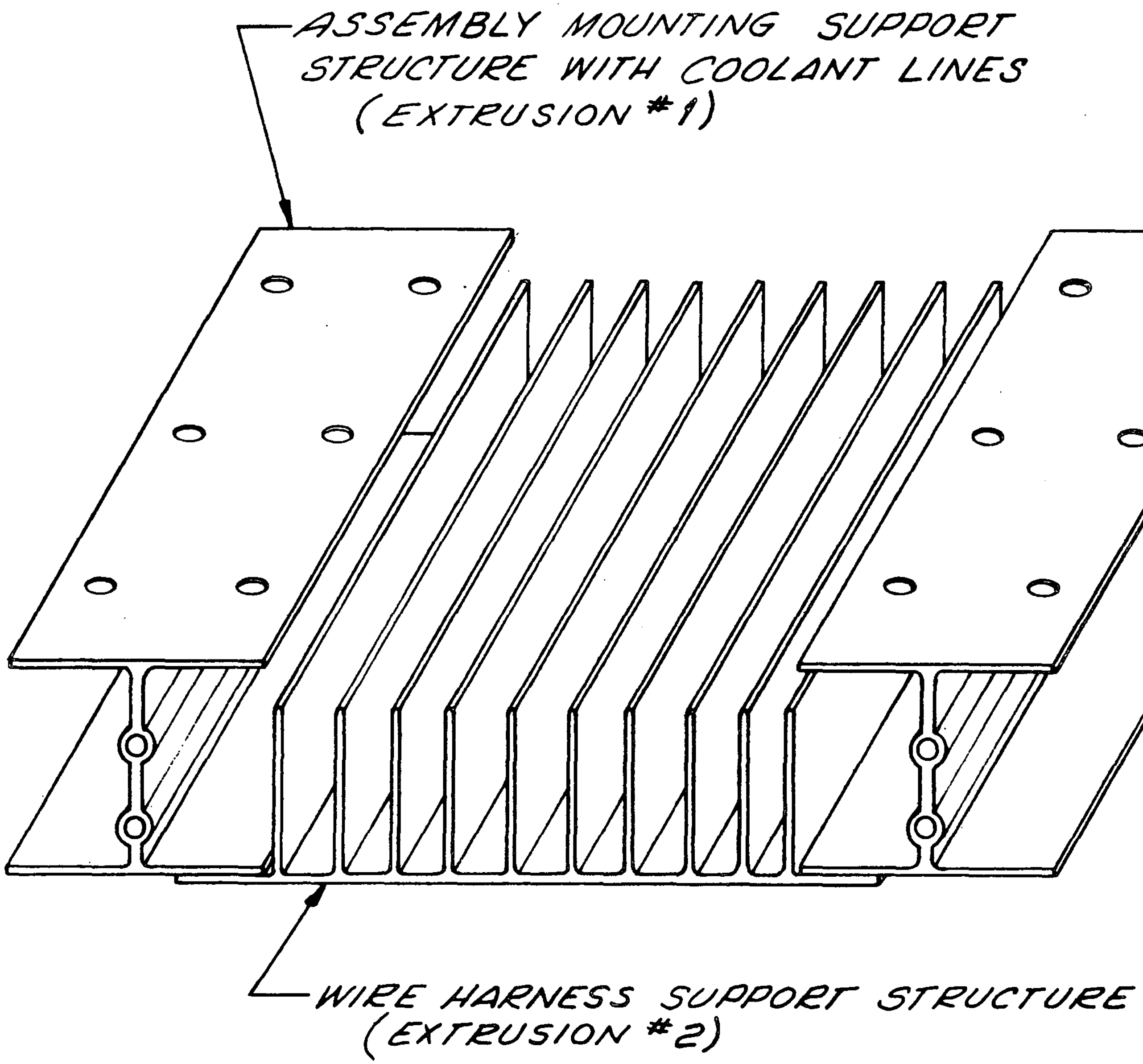

ASSEMBLY MOUNTING SUPPORT
STRUCTURE WITH COOLANT LINES
(EXTRUSION #1)
WIRE HARNESS SUPPORT STRUCTURE
(EXTRUSION #2)

Figure 6-10. Basic Structural Members

electronic assemblies and a wiring support channel extrusion containing longitudinal fins for wire routing and EMI separation. Since both of these elements are inexpensive extrusions, varying lengths can easily be used to fabricate different sizes of the installation section as required.

An integrated support structure is built up of a number of two basic element assemblies mounted side by side and bolted to top and bottom cross members, which in turn are mounted to the vehicle structural interface.

6.3.1.2 Wiring System

The wiring system consists of the integration of a series of wire harness runs installed in specific EMI category channels in the bottom of the basic fin-shaped wire support member. The wires, grouped according to a specified EMI category, egress from these channels up to the respective connectors on top of the replaceable assembly (see Figure 6-11).

With this wiring technique, management of the wiring complex is rigorously controlled since the routing in each wire channel is defined. Wire bundle size in many cases is minimized as wiring from the vehicle interface is divided between vertical rows of the replaceable assemblies. Where required, the interconnection of horizontal rows of replaceable assemblies is achieved in the horizontal space between assemblies.

The vehicle wiring interface is established in a wire channel at one or both ends of the structural complex. The wires egress from the fin shaped channels and are interconnected in the wiring cross channel in their respective EMI categories. The vehicle interface connectors are mounted on the bottom surface of the cross channel and connected to the appropriate wire harness.

Wiring costs should be minimized because wire harnesses would be prefabricated with minimal jigs or could be fabricated within the structural complex.

Round connectors with crimp-type pins and sealed rear insertion are contemplated at this time since they are well qualified. Rectangular connectors,

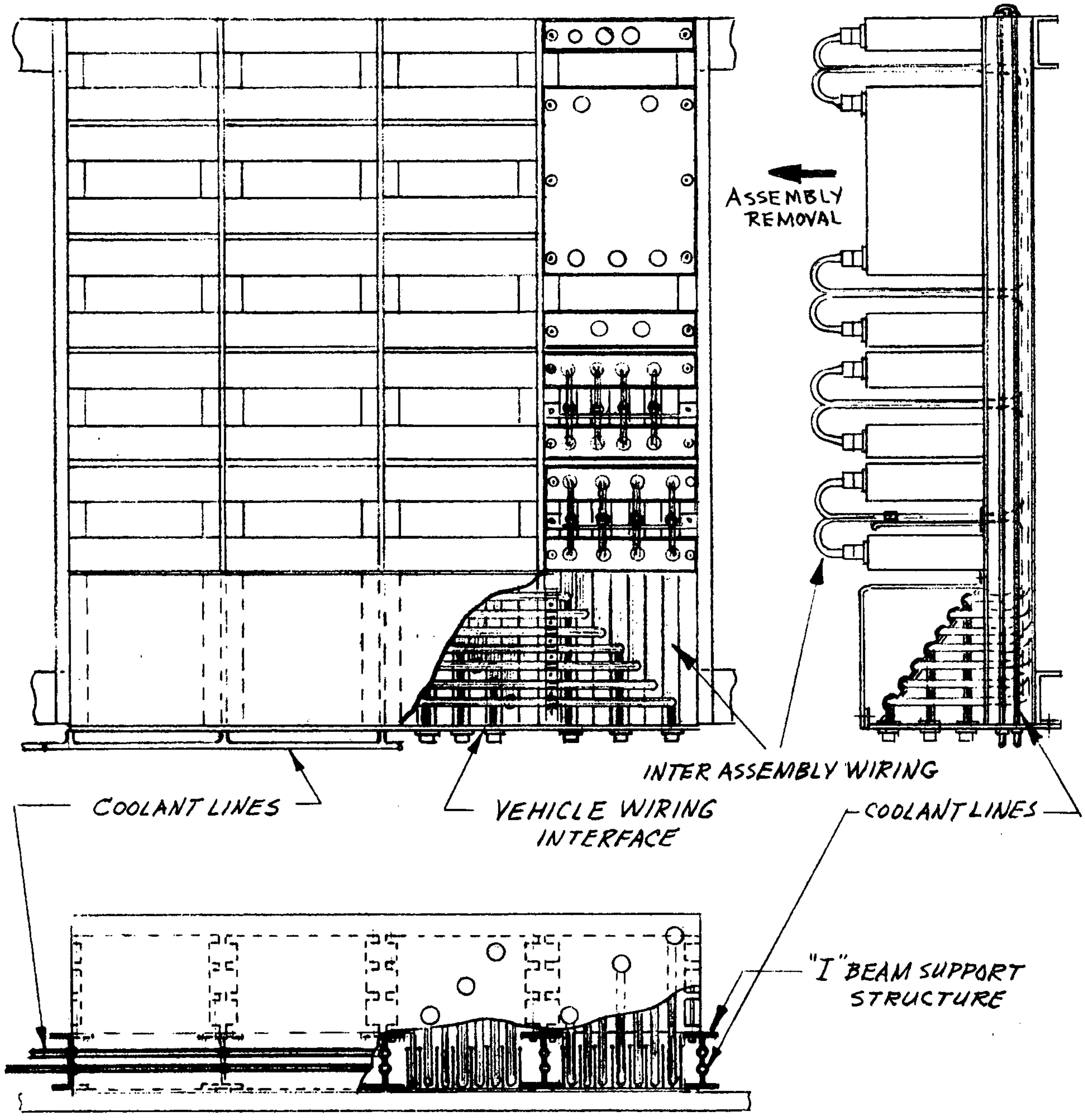


Figure 6-11. Installation Section-Wiring and Cooling Configuration

with some development, would offer certain advantages, particularly in space utilization.

6.3.1.3 Heat Transfer System

The heat transfer system consists of coolant lines integral to each of the basic "I" shaped structural member. This provides the capability of redundant coolant lines, which are interconnected independently in the cross channels at either end of the structural complex. The vehicle interface connections are located in these cross channels. This arrangement creates a short heat path between the coolant line and the mounting bolts of the replaceable assembly, thereby minimizing the thermal gradient across the interface.

Heat pipes could also be mounted to the sides of the "I" shaped structural member if this cooling technique proved to be more desirable. An investigation into use of heat pipes for cooling in lieu of a fluid flow cooling technique should be continued. Use of heat pipe techniques at the vehicle interface hinge line may provide an ideal interface that would allow disconnection (during a period when the equipment enclosure is required to be moved, such as for exterior wall inspection) without danger of loss of cooling fluid.

A heat-transfer capability of 20 watts per replaceable assembly could be achieved with a coolant-fluid temperature of 90° F.

In the event of a damaged coolant line, the respective side of the "I" shaped structural member could be easily removed and replaced without disrupting the entire installation complex.

6.3.1.4 Replaceable Assembly

The basic replaceable assembly has a form factor of 9 in. long by 7 in. high by 2 in. wide, as shown in Figure 6-12.

The assembly has the following basic features:

A. The assembly consists of a top and bottom plate, each bolted to two extruded side plates. The top plate contains the wiring interface

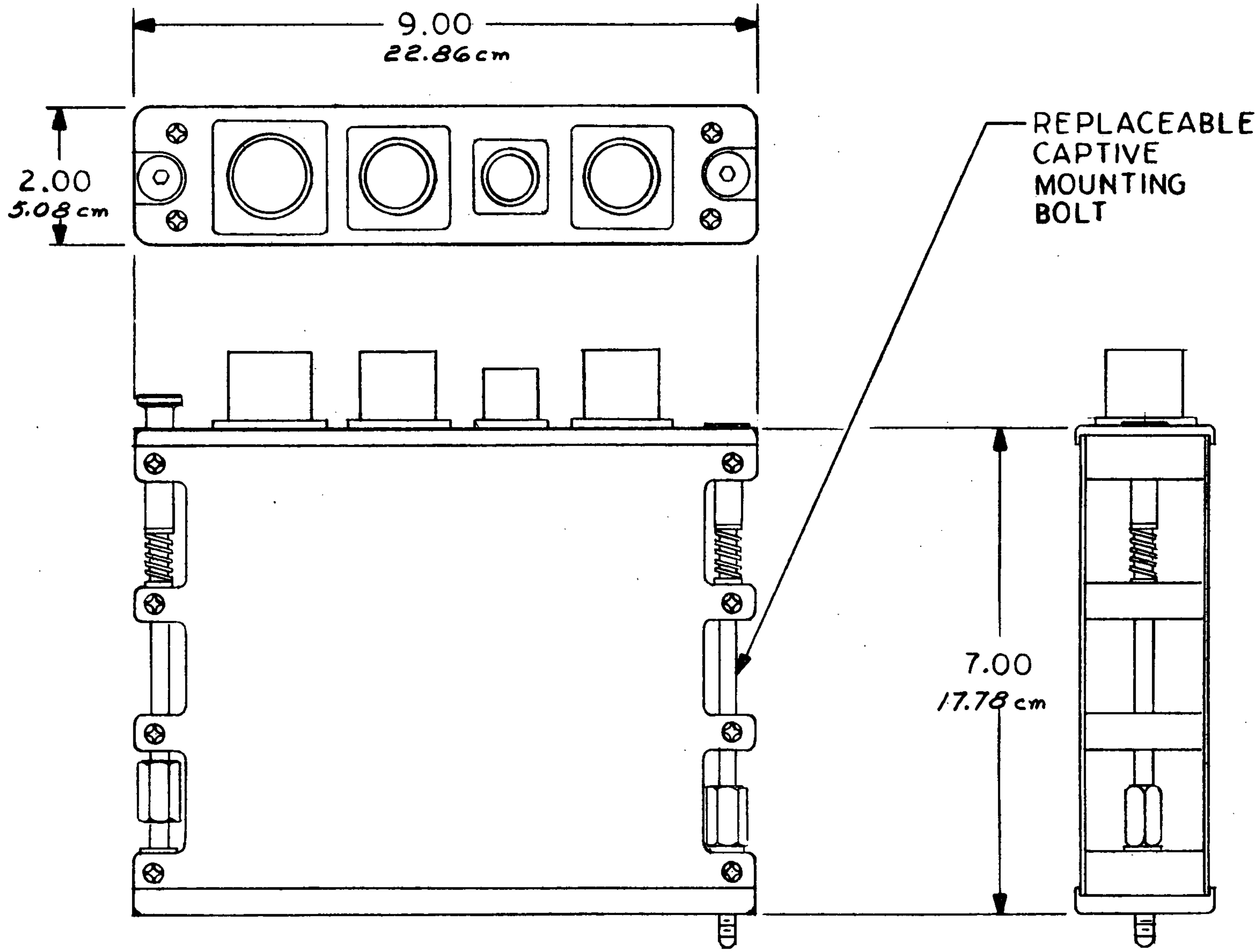


Figure 6-12. Basic Replaceable Electrical Assembly

connectors and mounting provisions for two 5- by 7-in. printed circuit boards containing about 150 discrete components per board. These boards could also be used for mounting thick film substrates, which would result in achieving much greater packaging densities. The bottom plate is used for mounting any high heat dissipating devices as required. The open sides of the assembled unit are covered to inhibit potential EMI or flammability problems.

B. The heat-transfer interface area around each bolt is about 2.50 sq in., or about 5 sq in. for each basic assembly. Each assembly has the capability of dissipating 20 watts.

C. The assembly has two captivated 0.25-inch-diameter mounting bolts, which are visually and physically accessible and also replaceable if damaged.

D. Keying of the assembly is accomplished by providing a family of pins with different locations in the "I" shaped structural element and mating holes in the replaceable assembly.

E. Multiple sizes within the family of the basic replaceable assembly are achieved as the side supports of the assembly are made from an extruded shape, which can be readily cut to any length.

F. The estimated weight for the basic replaceable assembly is 5 pounds.

G. Larger replaceable assemblies will retain the standard length (9 inches) and height (7 inches), the width may be increased in 2-inch increments.

Section 7
CHECKOUT LANGUAGE

The objective of this portion of the study was to develop the definition of a higher-order, test-engineering-oriented computer language suitable for the Space Station (and similar manned spacecraft). It is anticipated that this language will be utilized for the preparation of all Space Station checkout programs, both for preflight and on-orbit use. These test programs will for the most part be prepared on the ground in advance of actual use and stored for callup as needed. As will be pointed out, however, requirements also exist for on-line preparation and modification of procedures, and this function will utilize portions of the language. Further, it will be seen that the checkout language encompasses many characteristics that are pertinent to all automated onboard functions, including subsystem control and experiment operations. It is therefore postulated that one general and sufficiently flexible programming language could be used to create all DMS programs.

The definition of the checkout language was developed by first examining previous efforts in the design of such languages in an effort to benefit from them. Design requirements relative to the Space Station were then identified, and design objectives were formulated. The definition is the end product of these analyses.

7.1 LANGUAGE SURVEY

7.1.1 Languages Considered

The Automatic Checkout Language (ACL) survey considered the following, which represent a cross-section of the spectrum of ACL's that have been developed to date.

ATOLL (Acceptance, Test, or Launch Language)

This language is the result of a continuing design by the NASA Automation Committee, comprising representatives of NASA and the Apollo contractors, for ground checkout and prelaunch testing of Saturn launch vehicles. Although it falls short of state-of-the-art ACL design in several respects, ATOLL has been used actively for several years, probably substantially more than any other ACL. Any new design would experience problems in gaining acceptance if it did not at least incorporate the test programming features provided by ATOLL.

TOOL (Test-Oriented On-Board Language)
This language, developed by Martin Marietta, has features not available in ATOLL, these being provision for fully automated integrated multitesting, and options to make test programs more readable, although far from English sentence-like in structure. The language is highly specialized to a particular onboard checkout system (OCS). Stimulus and measurement parameters must be specified in detail in the program, as compared with being maintained in a "dictionary."

DMSOL (Data Management System — Operational Language)
This language, also known as OCS Test Language and OCS II, is a moderate departure from TOOL defined by General Electric. Its more significant differences make it lean toward ATOLL, in that fixed fields are used and highly abbreviated modifiers are required forms. DMSOL is the language adopted for use on the OCS breadboard at NASA MSC, and it is very specialized to that hardware. Test procedures are compiled and executed under control of a very powerful software operating system.

STOL (Saturn Test-Oriented Language)
This language was developed by MDAC (then Douglas) for the ground checkout of the SIVB stage, and has undergone very extensive use in that connection. STOL is a simple, English-like language that features the provision for reference to test equipment and vehicle points by English descriptive names as long as 40 characters. Specific information regarding these points was kept independent from the language, in a dictionary, for use at execution time.

MOLTOL (MOL Test Oriented Language)

This language was under development by MDAC for the Manned Orbiting Laboratory (MOL) program. It was based on the higher-level language PL/1, had capabilities for very general purposes, allowed declaration of and operations on a large variety of data types and structures, and incorporated STOL as a subset of its test language statements. MOLTOL's point of distinction lay in being the first test language designed for automated integrated or subsystem multitesting.

ATLAS (Abbreviated Test Language for Avionics Systems)

This language was developed by the Airlines Electronic Engineering Committee under the auspices of Aeronautical Radio, Inc. (ARINC). It is the classical example of a language designed for "bench testing," making extensive provision for the measurement of detail signal characteristics, as opposed to stage testing, where the values and presence or absence of signals are essentially the only characteristics of concern. Unhampered dissemination of procedures written in ATLAS was a chief objective, so inevitable variations of test equipment between using airlines' maintenance installations were made transparent by making test equipment nonaddressable except as it impinges on connector pins of the unit under test (UUT).

7.1.2 Fundamental Characteristics

The first systematic comparison and tabulation of properties of the above languages is presented in Table 7-1, Fundamental Characteristics of Language Surveyed. The abbreviations used have meanings as follows:

Opt.	-	optional
Req.	-	required
Opr.	-	by operator
Prg.	-	by program
TBD	-	capability intended, but not defined, in available documentation
Lim.	-	limited to one binary operation at a time
Mod.	-	moderate capability for formula evaluation
Ext.	-	extensive

The characteristics considered are defined as follows.

Table 7-1

FUNDAMENTAL CHARACTERISTICS OF LANGUAGES SURVEYED

Characteristic	ATOLL	TOOL	DMSOL	STOL	MOLTOL	ATLAS
English-sentence-like statements	No	No	No	Yes	Yes	Yes
Fixed-field statements	Yes	No	Yes	No	No	No
Abbreviated language vocabulary	None	Opt	Req	No	No	Req
English-like variable names	No	Opt	Opt	Yes	Yes	Yes
Coded internal variable names	Yes	Yes	Yes	No	No	No
Coded test point designations	Req	Opt	Opt	No	No	Req
Dictionary of instrumentation data	Yes	No	No	Yes	Yes	No
Multiprogramming capability	No	Yes	Yes	No	Yes	No
Monitoring (concurrent, software)	No	Yes	Yes	No	Yes	No
On-line data input capability	Opr	No	Prg	Opr	Both	No
On-line statement input and execution	No	No	TBD	Yes	Yes	No
On-line program change capability	No	Yes	Yes	Yes	Yes	No
Selective control of test progress	No	No	No	Yes	Yes	No
Bench testing suitability	No	Fair	Fair	No	No	Yes
Computational capabilities	Lim	Lim	Mod	Mod	Ext	TBD

7.1.2.1 English-Sentence-Like Statements

Everything to the right of the characters that identify the statement (name, "verb", operator, "element") is in variable field format, in phrases almost always introduced by "keywords" (TO, UNTIL, etc), which at once lend readability and identify the parameters of the phrase.

7.1.2.2 Fixed-Field Statements.

All source input lines have imaginary predefined breaks at the same character positions, regardless of statement type. At least one field is occupied to identify the statement. Different statements may very well use the same fixed field to contain parameters of different types. Sometimes a part of a statement is allowed to cross fixed-field boundaries, because of space limitations. Note that TOOL is not classed as being either of this or the above type, because its variable fields often contain only numeric values.

7.1.2.3 Abbreviated Language Vocabulary

The element names and alphabetic parameters have optional, but specific, abbreviations. The operators of ATOLL, the modifiers of DMSOL and some of the keywords of ATLAS have fixed forms, which are abbreviations of from two to four letters. A number of these abbreviations are not intrinsically meaningful, nor are they to industry-wide standards as are the abbreviations for engineering units.

7.1.2.4 English-Like Variable Names.

User-defined names can be used for variables, which can be as meaningful as the language's size limitation permits. TOOL, DMSOL, and ATLAS allow names of only seven, eight and sixteen characters long, respectively; and they must be defined in each program in which they are used. STOL and MOLTOL allow names 40 and 28 characters long, respectively, and all external function names are predefined in the dictionary.

7.1.2.5 Coded Internal Variable Names

A "yes" here means that all available internal variable cells have been assigned fixed symbols, such as DO-D9 or BSOO-BS99. DMSOL does allow the user to equate these to other symbols, although limited to eight characters. STOL, incidentally, uses coded names for time cells, TCxx. In any case, no satisfactorily meaningful identification of the content of such cells is provided for.

7.1.2.6 Coded Test Point Designations.

Test points are named according to a fixed format code, part of which may carry significance, but which never provides a full functional description of the signals involved. For example, DI127 could be a discrete input, but the number 127 does not impart any further information. TOOL, DMSOL, and ATLAS do allow these to be equated to user-defined names, but limited to 7, 8, and 16 characters, as mentioned above. In ATLAS, the user-defined name can refer to test points or any or all of the attributes of a signal to be measured or applied there.

7.1.2.7 Dictionary of Instrumentation Data

This is a file of information separate from all test program statements, which holds all of the data about each signal, checkout equipment accessible point, and vehicle test point that will be relevant at execution time for all of the commands, requests, and measurements in all of the test programs. Some of the kinds of information that might be included are precise definitions of wave shapes to be issued or expected, relay command codes, single-bit masks for multidiscrete inputs, the number of the assigned converter, and coefficients of the calibration equation. The file may be subdivided by subsystem or otherwise, but it is basically organized by name and data pertinent thereto. The alternatives to a dictionary are to include the information in test statements or build it into the operating system. Use of a dictionary data file removes dependence of the language upon particular hardware, makes changing of the data centralized and easier, and facilitates configuration control of both the programs and the equipment.

7.1.2.8 Multiprogramming Capability

For the most part, this is provided by the operating system, but statements are needed to enable the capability. In addition, MOLTOL has statements to synchronize concurrent programs where desired, and to selectively lock out interference.

7.1.2.9 Monitoring (Concurrent, Software)

ATOLL and STOL do not have or need this capability because the desired extent of concurrent equipment status monitoring is performed by hardware. However, software monitoring would be more economical and flexible. On the other hand, on a future manned space vehicle, the sheer volume of data that will need to be monitored to make long-term orbital periods feasible, among other considerations, leads to the belief that monitoring by a central computer will be concerned almost only with status words maintained by remote built-in or accessory test equipment (such as the Remote Data Acquisition Units on Space Station) which will perform the classical monitoring for out-of-limits conditions.

7.1.2.10 On-Line Data Input Capability

It may be desirable to allow an operator the ability to set or alter a carefully restricted set of internal test program parameters, while the program concerned is temporarily halted or otherwise active. A safer, more restricted capability is to let the program request data inputs, or advice, at certain points, where the operator inputs are not only screened for acceptable value by the program or operating system, but can only affect the one pre-programmed spectrum of alternative action that the input influences.

7.1.2.11 On-Line Statement Input and Execution

Particularly in a manned orbiting vehicle, great care must be taken in specifying the circumstances under which this capability would be allowed, if at all, and then what particular statements would be allowed. In a vastly safer environment, the on-line commands for the OCS breadboard at NASA MSC were the one element yet "to be determined" as of October 1970, the date of available documentation of the on-line, or "control" language associated with DMSOL. However, requesting of (standard) displays should always be provided for, although such displays must be displaceable by any preprogrammed need for the display space that may subsequently arise.

7.1.2.12 On-Line Program Change Capability

This, in the orbital situation, is the most potentially dangerous on-line capability. It should be ruled out absolutely for test programs affecting spacecraft operational subsystems, except possibly for alteration of parameters; i.e., on-line data input by the operator. With suitable interlocks, it might be acceptable to change experiment control programs aloft. On the ground, program changes (whether made on-line or not) should be subjected to verification procedures just as the original program was. The MOLTOL on-line modify mode permitted changes, providing they did not introduce any command that belonged to a command subset already allocated to some other active, concurrent test program.

7.1.2.13 Selective Control of Test Progress

By this is implied not just the alternative actions available at preprogrammed halts, but the capability of the operator to execute a program between preprogrammed safe pausing points, or to initiate a special procedure for backing up to a previous safe point. These features, provided by STOL and MOLTOL with the assistance of their operating systems, have been found very useful in factory and prelaunch checkout, although they have less apparent relevance in on-orbit operations.

7.1.2.14 Bench Testing Suitability

It is possible that some advanced manned space vehicle will have enough shop space to permit some on-orbit repair of modules whose faults have not been precisely identified, and which have been replaced by spares. In that case, separate bench testing programs might be employed, and ATLAS or some derivative would be a prime candidate for the source language of those programs. The UTEC (universal test equipment compiler), developed by U.S. Army Missile Command, is also recommended for consideration.

This kind of testing would defintely be conducted in ground maintenance depots. In any case, bench testing is out of the scope of the checkout language task undertaken. The analysis of the ATLAS language proved worthwhile, however, if only for the discovery of two of its features; namely, the TABLE data structure and programmed timing of the execution of future statements.

7.1.2.15 Computational Capabilities

Intended to be included here are any capabilities for manipulation of data, whether arithmetic or otherwise. In ATOLL and TOOL, this capability is limited to one elementary operation between two numbers per statement. In DMSOL and STOL, formulas containing elementary operations and parenthetical expressions can be evaluated, the value being "assigned" to a named variable. In MOLTOL, the formulas can be of almost arbitrary complexity, the array variables being allowed subscripts that can be of any permissible formula, the operations allowed including all the elementary functions, and any mixture of data types being allowed. Arguments can also be any permissible formula, and character and bit manipulation is provided for. Three special list-type data structures are defined, and special logical operations are supplied for them. Omitted, strangely, are any operations between arrays; e.g., matrix algebra.

7.1.3 Detail Characteristics

Table 7-2 presents characteristics of the languages surveyed at a considerably more-detailed level.

7.1.3.1 Fundamental Concepts

This lists the structural philosophy, nomenclature and restrictions if any for a single test program; the extent to which time-shared program execution can occur (as implemented by the languages' operating systems); the scope of existence (i.e., globality) of the various kinds of variables of each language; and how overlays are handled where information regarding this is known.

Table 7-2 (page 1 of 5)
DETAIL CHARACTERISTICS OF LANGUAGES SURVEYED

Characteristic	ATOLL	STOL	MOLTOL
Fundamental Concepts			
Program structure	Segmented program with nested ATOLL and assembly language subroutines, up to seven levels of ATOLL subroutines	Segmented program with nested STOL subroutines up to 10 levels, and imbedded assembly language as needed	Automatically segmented program of nested "blocks," simple, repetitive and "PROC's"
Multiprogram levels	None	None	Up to four, plus monitoring
Globality of variables	Globality within resident segment	Globality between resident segment and master control, current emergency stop and backup segments	Globality within a block, and all blocks nested inside only
Overlays	At end of segment execution, or due to transfer of control between segments	By master control at end of execution of ordinary segment; new emergency stop and backup segments overlaid when posted	At end of segment execution, or due to transfer of control between segments; nonresident PROC's overlaid when called

Characteristic	TOOL	DMSOL	ATLAS
Fundamental Concepts			
Program structure	Program ("task") consists of "test sequences" nested (by subroutine calls) to four levels	Task, of nested test sequences	Nested test procedures
Multiprogram levels	Up to four, including monitor	Up to eight, plus up to four monitors plus clock sequence plus control, plus assembly code routines up to memory limit	None
Globality of variables	Seven-character symbols, global within task; coded variables are universally global	Eight-character symbols global within task, as are 10 data registers. 100 buffer storage registers are universally global	16-character symbols, global within program defining them. 'MEASUREMENT' universally global, as are pin designations
Overlays	Complete task is assumed resident at start of execution	Complete task must be made resident prior to start of execution	Complete test program is assumed resident at start of execution

Characteristic	ATOL	STOL	MOLTOL
Character Set			
Upper-case letters	Yes	Yes	Yes
Digits 0-9	Yes	Yes	Yes

Table 7-2 (page 2 of 5)
DETAIL CHARACTERISTICS OF LANGUAGES SURVEYED

Characteristic	ATOLL	STOL	MOLTOL
Character Set (Continued)			
FORTRAN special set	Yes	Yes, less $	Yes
Others	No	No	No
Statement Format			
Control characters	None	H, S, N, E, or D in column 1	H, S, N, E, F, T, P, Q, M, R, or B in column 1
Comments	Columns 2-72 if * is in column 1	Columns 10-72 if * is in column 1	Columns 2-72 if * is in column 1
Continuation	Yes, if last character in variable field is a comma	Yes, if a digit 1 to 9 is in column 8	Yes, if C is in column 1
Time field	Execution constraint, or delay, in columns 35-42	No	Execution constraint, in columns 3-10
Label	Yes: 6-digit step, substep in columns 1-6	Yes, if alphanumerics are in columns 3-6	Yes, can be alphanumerics plus non-lead ".", in columns 12-17
Fixed fields	Operator, columns 7-10 Condition, column 11 Value, columns 12-19 Low limit, columns 20-25 High limit, columns 26-31 Units, columns 32-34 Time, columns 35-42	Operator, starts in column 10	Operator, starts in column 19
Variable field	Columns 43-70	Everything after operator	Everything after operator

Characteristic	TOOL	DMSOL	ATLAS
Character Set			
Upper-case letters	Yes	Yes	Yes
Digits 0-9	Yes	Yes	Yes
FORTRAN special	Yes	Yes	Yes
Others	No	No	Single quote (apostrophe)

Table 7-2 (page 3 of 5)
DETAIL CHARACTERISTICS OF LANGUAGES SURVEYED

Characteristic	TOOL	DMSOL	ATLAS
Statement Format			
Control characters	None	None	B, C, E, M, or S in first field
Comments	No	Yes, by REMARK statement	Yes, following B or C control character
Continuation	No	No	Yes, until $ sign is encountered
Time field	No	No	No
Label	Yes, if digits are in columns 1 or 2	Yes, if alphanumerics are in columns 1-9	Yes, second field, six or two digits
Fixed fields	"Element" (operation or avvreviation), columns 4-13	Element, columns 10-19 Mod, columns 20-23 DR, columns 24-26 Field "n", columns 17 + 10n to 26 + 10n	"Verb", third field
Variable field	"Modifiers," columns 15 on	No	Yes, everything following verb, often with subfields
Constants			
Integer (decimal)	Fixed-point integer, in context only	Fixed-point integer, in context only	Fixed-point integer, without ".".
Mixed number (decimal)	Fixed-point mixed number, 14 fractional bits, "." optional	Fixed-point mixed numbers, 10 fractional bits, "." optional	Fixed-point mixed number with ".", fractional bits as DECLARED
Octal	Distinguished by leading zero	Distinguished by being unsigned	Distinguished by trailing alpha "0"
Hexadecimal	No	No	No
Binary	No	No	Followed by "B"
Time	Delay, in milliseconds, or count-down time constraint	Mixed units optional, delay or range time constraint	Mixed units optional, delay or count-down time constraint
Textual	In display page image	Preceded by "nH/", n = number of characters	Enclosed by $ signs
Logical or Boolean	0 or 1 in Condition	In conditional phrases: 0 or 1, or certain words	In conditional phrases, same as STOL plus more word forms; in formulas, as octal constants
Multivalued arrays	"Discrete allow" tables	Constructed by TABLE statement; most types allowed, mixed	Constructed by DATA statement; only one type per array

Table 7-2 (page 4 of 5)
DETAIL CHARACTERISTICS OF LANGUAGES SURVEYED

Characteristic	TOOL	DMSOL	ATLAS
Constants			
Integer (decimal)	Fixed point, lead I, parentheses enclosed	All assumed floating point, E notation optional	Fixed or floating point; latter uses E notation
Mixed number (decimal)	Floating point, lead F, parentheses enclosed		
Octal	No	No	Yes, lead 0, quote enclosed
Hexadecimal	Yes, lead X, parentheses enclosed	Yes, in context	Yes, lead X, quote enclosed
Binary	No	No	Yes, lead B, quote enclosed
Time	Yes, in context	Yes, in context	Yes, in context and with units
Textual	Yes, enclosed by $ signs	Yes, enclosed by $ signs	Yes, lead C, quote enclosed
Logical or Boolean	In conditional phrases, as words	In conditional phrases, as words	In conditional phrases, as words
Multivalued arrays	No	In formulas as hex constants	Yes (in TABLE of PERFORM statement, only)

Characteristic	ATOLL	STOL	MOLTOL
Variables			
Symbolic name size limit and naming conventions	Coded designators of various sizes; 1 to 6 lead significant letters	40 alphanumerics, plus imbedded blanks. Lead letter required	28 alphanumerics plus optional imbedded periods. Lead letter required.
Internal variable types	All fixed point	All fixed point	All fixed point
Integer	Index variables X1-X7, flags Fnn	Index variables: I, J, K, L, M, and N only	Any name
Mixed number	Arithmetic AC1-AC63, analog test TAB0-TAB9; 14 fractional bits	Any unreserved name, 10 fractional bits	Any name, fractional bits as DECLARED
Logical, Boolean, or discrete	None	None	Any name, values 0 or 1

Table 7-2 (page 5 of 5)

DETAIL CHARACTERISTICS OF LANGUAGES SURVEYED

Characteristic	ATOLL	STOLL	MOLTOL
Variables (Continued)			
Internal variable types (Continued)			
Digital (Multidiscrete)	"Discrete allow profiles," one for each of six types	None	Any name, up to 24 bits, each bit addressable
Time	TCA-TCD and CDA-CDC, one word each, can hold GMT or CDC, respectively, in seconds	TCI-TC50, two words each, hold range time in milliseconds	Any name, two words each, hold computed times or values of global variables GMT, CDC, VC, or TIME, in milliseconds
Textual	None	Any unreserved name, any length, four characters per word	Any name, one word each, holds up to four characters
External variables	Various coded designators with leading letters according to type	Any name; represented by 1-word code after compilation	Any name. Represented by 1-word dictionary reference after compilation
Engineering units	Optional, ignored if given	Required but ignored	Optional; ignored if given
Multielement variables	None except above groups	1-dimensional arrays, any content	1, 2, or 3-dimensional array "lists"; content only as DECLARED for each

Characteristic	TOOL	DMSOL	ATLAS
Variables			
Symbolic name size limit and naming conventions	Seven-character alphanumeric, for externals only	Eight-character alphanumeric for externals and constants only	Alphanumeric, first 16 characters unique
Internal variable types	10 "data cells", with coded names D0-D9, could contain any of the internal variable types. Only mixed numbers are floating point	10 data cells with coded names D0-D9, plus 100 buffer storage registers, coded BS00-BS+99; could contain any of the internal variables types. Numbers are floating point	Any legitimate name or the global variable MEASUREMENT; could contain the value of any engineering quantity, or a computed function; i. e., in essence, mixed numbers. Always enclosed by quotes or parentheses, depending on context. Fixed-to-floating and size are implementation-dependent
Integer Mixed number			
Logical, Boolean, or discrete			
Digital (Multidiscrete)			
Time	(Two words)		
Textual	None	None	None
External variable	Symbolic or eight-character coded name if hardwire. Item number for PCM	Symbolic, or eight-character coded name for PCM or hardwire	Coded pin designations or symbolic names
Engineering units	Limited but relevant	No; modifiers imply fixed set of units	Large standard set, required for any measurement or stimulus
Multielement variables	None except data cells	None except above groups	None, as of supplement 1, version of specification

7.1.3.2 Character Set

This lists, in essence, the deviation (if any) from the standard set of FORTRAN IV. Not surprisingly, there are only two differences. The FORTRAN special characters are + - * / = . , ( ) $ blank. Many more special characters are available on most makes of keyboard today, but the advantages of using any of them must be balanced against the possible burden on the memories of users and loss of compatibility with the older keyboards, many of which are still in extensive use.

7.1.3.3 Statement Format

This lists all the kinds of fields found in the formats of all the languages, where they occur in the statements of each language, if at all, and what the permissible contents of the special fields are, if used. All of the formats are given in terms of source input as if from cards, except ATLAS, which assumes a free-form appearance to allow for serial input (such as from paper tape) where the order of fields is prescribed, and the content of a used field identifies its presence. MOLTOL is also like this by specification, but in practice, as with ATLAS, a coding form with fixed special fields was adopted.

7.1.3.4 Constants

This lists all of the types of fixed character strings encountered in all of the languages, thus comprising all constants, purely numeric and otherwise. For the languages that allow them, rules for forming the various constants are given. In no language can all kinds of constants (even numeric) appear wherever some kind can appear. The exceptions would have to be given statement-by-statement, and this will not be done. It seems worthwhile simply to take note of great variation in usage rules for constants, and resolve to improve on this situation in future designs.

7.1.3.5 Variables

This first gives the naming conventions and variable name-size limits, and then lists all of the types of computer-internal single-element variable types encountered in the survey, followed by information regarding external variable names and engineering units. Variables of a language may all be named according to the naming rules first given, except as noted. Limitations on the number of variable storage cells available are given as pertinent. All single-element internal variables require only one storage cell each, except as noted.

"External variable" means the name of a function or point of the test equipment or article under test. Note that the value, state, or other characteristic of the function or point is not implied by its name, except that in MOLTOL the mention of a measurable external function in a formula does cause the reading of its value, which is then utilized in evaluation of the formula. If values of external variables are to be "remembered," they must be kept in internal variables. (ATLAS automatically saves the *last* value read in the global variable MEASUREMENT.) Other characteristics of external variables are either written explicitly in the appropriate language statement or kept in the instrumentation data dictionary. The characteristic "engineering units," where allowed, may be optional; even if required, they may be ignored. If relevant, the operating system uses that information in the conversion process between external digital data and an internal value in engineering units.

7.1.3.6 Multi-Element Variables

This lists all of the types of internal variable data structures encountered that allow definition to include more than one element or value under a single overall name. The type called "arrays" is the same as that in FORTRAN. STOL array subscripts may be no more complex than an index variable plus or minus a digit, while MOLTOL array subscripts may be any evaluable expression. The rules for subscripting and using MOLTOL lists are quite extensive and will not be given here. The salient feature of these lists is probably that mention of the name of a list, unsubscripted, causes the contextual operation to be applied to all members of the list, just as in FORTRAN all elements of an array can be input or output by mentioning the unsubscripted name of the array in an appropriate statement.

7.1.4 Statements Provided

Table 7-3 is a side-by-side compendium of the operators (statement names, "verbs") found in the languages surveyed, grouped by statement categories, with those on the same row having at least a clear similarily of function. However, some equivalent statements have fairly substantial differences in detail, which for space considerations are not presented. The table does serve to demonstrate the entrapment of the English language, as in some cases each language uses a different synonym for the same function, while in other cases the same word is the operator for different functions; note the use of both END and RETURN for both the physical and logical conclusion of a subprogram, for example.

In some cases, again because of space considerations, the word shown is a modifier to be used in the generic statement of the concerned category, in order to create a statement functionally equivalent to those using the operator names given on the same row. Blank entries indicate no capability, according to the latest available documentation.

7.1.5 Strengths and Weaknesses

A critical analysis of the language characteristics tabulated above leads to the identification of the following advantages and disadvantages exhibited by the languages surveyed.

ADVANTAGES

English-sentence-like statements
Abbreviations optional except standard
Engineering function names allowed
Dictionary of instrumentation data
Computational capability for checkout
Capability for multiple actions via tables
Provision for multi-integrated testing

Table 7-3 (page 1 of 4)

OPERATOR EQUIVALENCES OF LANGUAGES SURVEYED

Statement Types	ATOLL	TOOL	DMSOL	STOL	MOLTOL	ATLAS
Program Structure	NAME	BEGIN	START	IDENT	BEGIN	BEGIN
	END	END	END	END	END	FINISH
			TERMINATE	FINIS		TERMINATE
			ENTRY	E	E	E
	BLOK			SEGMENT		
	MLSR			CALL$^{(E)}$	INCORP$^{(C)}$	
Subprogram Statements	BEGN	BEGIN	START	BEGIN	PROC	PROCEDURE
	END	END	END	RETURN	END	END
	EXEC	CALL	CALL	EXECUTE	EXECUTE	PERFORM
	RETN	END	RETURN	RETURN	RETURN	END
Assembly Code Insertion	MLSR$^{(E)}$			ENTER$^{(C)}$	ENTER$^{(C)}$	LEAVE$^{(C)}$
				ENTER	END	RESUME
	EXEM		EXECUTE			
Measurement	READ	MEASURE	MEASURE	MEASURE	READ	MEASURE
	READ	MEASURE	MEASURE		SAMPLE	MONITOR
	RGMT, RCDC, SETT	READ	READ	READ TIME	GMT, CDC, VC, TIME	MEASURE
					Implicit	
	PROB, PROC, PREM, SCAN					LOGIC SIGNAL
Condition Testing	States	States	States	States	States	States
	Limits	Limits	Limits	Limits	Limits	Limits
	Relations	Relations	Relations	Relations	Relations	Relations
	TEST, TFLG	CHECK	COMPARE	IF	IF	COMPARE, VERIFY

Table 7-3 (page 2 of 4)

OPERATOR EQUIVALENCES OF LANGUAGES SURVEYED

Statement Types	ATOLL	TOOL	DMSOL	STOL	MOLTOL	ATLAS
Timing Controls	DELY	DELAY	DELAY	DELAY	DELAY	DELAY
	DELY		WAIT FOR	WAIT	WAIT	WAIT FOR
				WAIT UNLESS	TIME SINCE	WAIT FOR STEP
				N	N, IMMED	
	TEST				Time field	
		INTERRUPT	INTERRUPT			
Monitoring	MNTR	START	LOAD, SCAN	POST SIM	MONITOR	
	MNTR	STOP	STOP	POST SIM	RELEASE	
			EDITBUF			
Stimulation		STIMULATE	STIMULATE	APPLY	APPLY	APPLY
	DISO, SSEL	CONNECT	SET	SET	SET	CLOSE
	DISO, SSEL	DISCONNECT	CLR	RESET	RESET	OPEN
	MDSO				SET list	LOGIC SIGNAL
	DISO	PULSE	VPL		PULSE	LOGIC SIGNAL
		DIGITAL		SEND	APPLY	
		TRANSFER			DISPATCH	
		CLEAR				REMOVE
	DISA				INHIBIT	
	DISA				RELEASE	
						ADJUST PREPARE-EXEC.
Multiprogramming Control		START	LOAD		START	
		STOP	STOP		END	
					SYNC	

Table 7-3 (page 3 of 4)

OPERATOR EQUIVALENCES OF LANGUAGES SURVEYED

Statement Types	ATOLL	TOOL	DMSOL	STOL	MOLTOL	ATLAS
Operator Controls Related				POST BACKUP	POST BACKUP	
	TERM			POST EMERGENCY STOP	POST EMERGENCY STOP	
				H, S	H, S, B	
		REPEAT				
Programmable Sequence Control	TEST	IF	COMPARE	IF	IF	IF
	GOTO	GOTO	GOTO	GO TO	GO TO	GO TO
	SETX	DO		INDEX	DO	ALTER
	INCX	AGAIN		REPEAT	END	REPEAT
Execution Pause Control	SEMI			HALT	HALT	
			INPUT		REQUEST	OPERATOR INTERVENTION
On-Line Operations			TBD	EXECUTE	EXECUTE	
			REVIEW	CHANGE	MODIFY	
			COMPILE			
	TPRQ		LOAD	START	START	
	$PT		STOP	STOP	STOP	
	$PX1*		DISPLAY	DISPLAY	START	
	$PA		CALL	GO TO	GO TO	
	$PO			COAL	Assignment	
Readouts	DPLY	DISPLAY	DISPLAY	DISPLAY	DISPLAY	DISPLAY
	RECD			PRINT	PRINT	PRINT
	DFLG	MILESTONE		INDICATE	DISPLAY FORMAT	INDICATE

Table 7-3 (page 4 of 4)
OPERATOR EQUIVALENCES OF LANGUAGES SURVEYED

Statement Types	ATOLL	TOOL	DMSOL	STOL	MOLTOL	ATLAS
Peripheral Input/Output	RECD	RECORD	RECORD		TAPE	RECORD
			INPUT		DATA TO	
			RETRIEVE			
Variable Definition	DECL			DEFINE	DECLARE	DEFINE
				TABLE	DATA	TABLE
	LDAC	Assignment	Assignment	LOAD	Assignment	
			EQUATE		DEFINE	SPECIFY
				RESERVE	ARRAY	
Computational Capability	Binary operations	Binary operations	Expressions	Expressions with Subscripts	Expressions with Expression Subscripts	TBD (CALCULATE)
	Arithmetic assignment (ARTH)	Arithmetic assignment (EVALUATE)	Arithmetic assignment (SOLVE)	Arithmetic assignment (COMPUTE)	Arithmetic assigment	
	Flag setting (SFLG)	Logical assignment (EVALUATE)	Logical assignment (SOLVE)		Logical assignment	
				Time assignment (COMPUTE)	Time assignment	
					List assignment	

NOTE: (E) – execution time load; (C) – compile time incorporation

DISADVANTAGES

Fixed field format
Required nonuniversal abbreviations
Coded names of variables required
Hardware dependence in language
Excessive manipulative capabilities
Only single-action checkout statements
Only one execution sequence at a time

None of the languages surveyed or otherwise known of possesses all of the above advantages, and in some cases the degree to which a language can claim an advantage is not considered adequate; e.g., names of engineering functions that are limited to seven or eight characters simply cannot be explicitly meaningful except in rare cases.

It is not pertinent nor possible to "grade" the languages surveyed, because of overriding considerations that are not the concern of a task that attempts to define what a future checkout language (that for the Space Station) should be like. For example, ATOLL has most of the disadvantages listed, as well as others not shown. This would be generally agreed, yet ATOLL is an extremely viable checkout language today, because of the investment in training and accumulation of experience over many years. Furthermore, ATOLL has a limited but invaluable form of "capability for multiple actions via tables" in its statements dealing with "profiles" and "discrete-allow" tables.

Some specific objections to fixed-field format are as follows:

A. Straightforward listings of (ATOLL) program cards are unreasonably hard to read because there are no spaces between the fields.

B. The necessarily small fields lead to unacceptably brief abbreviations.

C. The potential advantage of fixed-field format (namely, fields with dedicated uses so labeled) is largely lost to ATOLL and DMSOL, because the large number of different items that may appear in the statements as a whole force the languages to use a given field for different items in different statements, or to use a shortened variable field anyway, as a catch-all. This leads to column labels as vague as "Value," "Variable," "Modifier," or "Field 1," "Field 2," etc.

It seems reasonable that universally accepted abbreviations for engineering units should be used, but in the interests of readability and comprehensibility with absolute minimal training, any other use of abbreviations should either be disallowed or optional.

For the same reasons, it should at least be optional to apply engineering-English names for test points, instrumentation, and computer-internal data. An option is proposed for output listings whereby all names would appear either in their coded form, or in their word-described form.

There will not be unanimity about the desirability of a dictionary, because there is some resulting loss of immediate visibility of the detail characteristics of stimuli and measurements in the statements naming them. However, listings of these details can be provided. The insulation of checkout programs from hardware changes, and the possibilities for automated, tight configuration control that a dictionary provides, are felt to be a very worthwhile exchange for the nuisance of coding of hardware information in checkout programs.

MOLTOL's main weakness, paradoxically, lies in the strength of its capabilities for manipulating data, that are far beyond the needs of almost all test engineers. As will be proposed in the language design presented here, where the need for a very large range of a given type of capability is anticipated, considering all users, the capability should be partitioned and allocated to language subsets. Individual users would only be taught relevant subsets, such that the training and acceptance problems would be greatly eased, overall.

As already mentioned, the "capability for multiple actions via tables" seems very desirable, and is found in the proposed language in a form extended and generalized from ATOLL.

Capability for concurrent subsystem testing and integrated systems testing seems a must, particularly in the autonomous, potentially time-critical environment of the Space Station.

7.2 FUNCTIONAL REQUIREMENTS

The onboard checkout system (and thus the OCS language) is in effect a tool, interfacing with the crew on one side and with the subsystems on the other. Requirements for these interfaces and some general considerations are discussed below.

7.2.1 General

A number of general considerations are applicable to the design of a checkout language such as the skill of the user and the nature of the application. These are discussed in the following sections.

7.2.1.1 User-Skill Category

Experience indicates that the most effective development of checkout programs results from providing knowledgeable, responsible engineers with a sufficiently high-level, problem-oriented language to describe the requisite test procedures. It is therefore intended that the language shall be effectively usable by design, test, and flight engineers and system specialists having no prior programming experience, and after only a very few days of training. It shall therefore be possible for programmers to use the language effectively also, provided they are given sufficiently detailed testing requirements or thorough indoctrination in the subsystems to be tested.

Regardless of who performs checkout program development, there is a need for the language compiler output listing to be readable and meaningful to crew, engineering, quality assurance, and other personnel. This permits direct utilization of the compiler output as a printed procedure for reference, configuration control, selloff, and other purposes requiring readable documentation.

7.2.1.2 Usage Location

It is assumed that the great majority of checkout program development will be accomplished on the ground, even after the Space Station is on orbit. It is projected, however, that the need will exist for onboard personnel to be able to modify or add to the checkout program repertoire, possibly because of time urgency or simple expediency in getting a job done. This usage can be conveniently made at the interactive controls and display consoles that will exist for spacecraft and experiment subsystems control. For convenience and minimization of training requirements, it would be highly desirable for the checkout programs on the ground to be developed at the same (type of) consoles, which will undoubtedly exist anyway for preflight training of all onboard personnel.

7.2.1.3 Level of Testing

To meet the objective of the onboard checkout system (OCS), the Space Station checkout language must have capabilities for fault isolation to a line replaceable unit (LRU) in any circumstances where this is meaningful and feasible from the viewpoint of instrumentation. This translates into requirements for substantive logical analysis and data handling capability on the part of the software. Virtually all forms of Space Station data can be involved, including subsystem and experiment data, crew interfaces, and internal DMS data.

In addition, the OCS must have capability for accommodating higher levels of checkout, including integrated system testing, sometimes in parallel with other operations. This imposes requirements on the checkout language for orderly subsystem checkout program execution, concurrent with other checkout or operational programs.

7.2.1.4 Mission Phase

While the capabilities of the checkout language are primarily intended to suit it for development of programs for orbital checkout, the result is a language that may also be used in preflight and ground-support operations involving testing down to the LRU level and up to integrated systems level. Such testing could take place at fabrication as well as launch sites, and would be included in prelaunch operations concerning Space Station modules of all types.

It is considered that the Space Station checkout language must be able to support the checkout requirements of experiment subsystems, although these are far from being precisely defined at this time. It can be safely assumed that telemetry will be involved (e.g., for free-flying modules) and that calibration will be required for a wide variety of sensor systems, including imaging. Experiment checkout may require the specific positioning of experiment modules or of the Space Station, implying the need for coordinated cooperation of GNC functions. Finally, the Space Station may provide interim checkout capabilities to docked logistics vehicles.

7.2.2 Subsystem Checkout

The following subsystems are identified as requiring checkout support:

A. Guidance, navigation, and control
B. Environmental control and life support
C. Communications
D. High-thrust propulsion
E. Low-thrust propulsion
F. Data management
G. Structure
H. Docking provisions
I. Experiments

7.2.2.1 Signal Types

Data inputs and data outputs that the OCS will make concerning subsystems are signals defined as measurements and stimuli, respectively. Both of these signal types find further subdivision into analog, bilevel (discrete), and digital (multi-digit binary codes or integers). Of course, analog signals as seen by the computer are also digital (conversion having taken place externally), the main difference being that the digitized analog values are subject to possible need for scaling and calibration. Analytic or format conversion may be necessary at the input-output interface for any signal type.

The OCS programs may be concerned with at least two other signal types, but indirectly in essence. RF stimuli can be generated in the communications subsystem when appropriate, via remote discrete control by the computer. RF measurements are reduced to bilevel or digitized analog forms, where needed by the computer. Pulse stimuli (i.e., discretes held for specified periods of time, possibly repeated periodically) can be handled in the same way as RF stimuli, but in the case of single pulses of moderate to long-duration the language can easily supply the capability to issue these. Pulse measurements will be reduced by subsystem instrumentation to digital values, e.g., pulse durations, pulse repetition rates, etc.

7.2.2.2 Signal Characteristics

Following are listed typical LRU level devices along with representative measurements associated with the performance. While not exhaustive, the list is indicative of the concerns and the conversion problems of the checkout programs:

Devices	Measurements
Gas storage vessels	pressure, temperature
Valves	position
Heaters	temperature, current
Accumulators	pressure
Pumps	rotation speed, pressure differential
Filter beds	temperature, humidity, pressure
Urine purification assemblies	flow, ph, temperature
Solar panels	current, voltage, temperature
Batteries	voltage, temperature
Relays, contactors	position, coil voltage
Circuit protection devices	test current,position
Hatch seals	pressure
Transmitters	RF power, frequency
Receivers	AGC level, output level
Gyros	temperature, rebalance commands, output signals
Star tracker	gimbal position, dissector output
CMGs	spin rate, vibration, bearing temperature

7.2.2.3 Checkout Functions

The following checkout functions have been identified:

A. operational status

B. caution

C. warning

D. periodic

E. trend analysis

F. fault isolation

G. reconfiguration

H. calibration/re-certification.

The term "status monitoring" includes the functions of operational status plus caution and warning, the essential difference being the seriousness of a fault if detected; i.e., imminent danger to crew, imminent threat to successful mission completion, or imminent likelihood of degraded subsystem performance.

Periodic testing may be thought of as more thorough or in-depth status checking. It may involve onboard personnel in two ways: on-demand initiation of specific modules of the periodic testing software, or crew observation of subsystem performance in concert with computerized evaluation.

Fault isolation has the objective, as discussed earlier, to isolate a detected malfunction to the LRU causing it.

Reconfiguration involves automated restoration of capability, plus bookkeeping in case of manual replacements by onboard personnel. Calibration or recertification is required for manually replaced LRU's. (Redundant LRU's that are automatically switched in by reconfiguration are certified as a normal part of periodic checkout operations.)

7.2.2.4 Checkout Strategy

The basically simple pattern of automated checkout that will be followed in every subsystem is as follows:

A. Verify operational stimuli, or apply one or more initial standard stimuli having removed operational stimuli, if any.

B. If any standard stimuli were applied, wait an appropriate length of time for the subsystem to respond as expected.

C. Make one or more measurements.

D. Compare (a transformation of) the measurements with one or more (sets of) standards or limits.

E. If the comparison is favorable, proceed to another checkout task, observing scheduling constraints, if any. If not favorable, use an appropriate means to resolve the situation. As soon as appropriate, replace or remove the standard stimuli, if any. If not potentially dangerous, problem resolution should proceed concurrent with further checkout.

The fault disposition of Step E may be accomplished automatically or by crew-assisted means. The former may involve issuance of additional stimuli or commands, additional measurements, hardware or software reconfiguration, and crew notification by displays. The latter may involve all of the above also, and certainly will require substantive use of a means for communication with the crew to direct crew activity or to allow crew direction of further computer activity.

In any case, control inputs from the crew may have to be acted on at any time, some of which may alter the scheduled sequence of the checkout operations.

7.2.3 Man-Machine Interface

Inasmuch as the Space Station is a manned vehicle, with the crewmen being an essential element of the onboard checkout capability, the requirements associated with the man-machine interface are important. These are discussed in the following paragraphs.

7.2.3.1 Interactive Requirements

An effective method for developing the characteristics and thus the requirements of a man-machine interface is to start with the absolutely fundamental nature of all such interfaces, and expand upon it in light of the particular interface of interest.

A generalized man-machine interface may be defined to be an environment in which both the man and the machine can issue commands, requests for information, and gratuitous information to the other. Recognizing that these are really requests for action, requests for information, and requests to accept information, respectively, it is seen that a man-machine interface is basically an arrangement for the intertransmittal of requests. One or more forms of meaningful human communication, usually including visual displays, are indispensable adjuncts to the request interface, for machine acknowledgements and responses to the requests made by man, and for presentation of the requests made by the machine. Thus evolves the rigorous logical basis for providing a man-machine interface with controls and displays.

Corollary observations are that some displays, on a dedicated, time-shared, or space-allocation basis, can serve as controls on man; and that, while all of the actions of the machine are in the final analysis due to man's creation of them (through programming), some portions of the machine's actions are preplanned to be triggered by mechanistic external events, including the elapse of prespecified times, while the remaining portion should properly only occur in response to real-time requests by man. The former actions will be termed "computer-initiated" and the latter "man-initiated."

It is anticipated that onboard checkout of the Space Station will normally operate substantially in an automatic preplanned manner. Thus, in developing the requirements for the man-machine interface on the Space Station, special consideration must be given to those transactions that are computer-initiated.

7.2.3.2 Controls and Displays

The following checkout-related requirements for an interactive man-machine interface are identified:

A. Provide for orderly initiation and control of standard procedures.

B. At decision points, display available alternatives and supporting data as pertinent.

C. Provide for data entry, as required by the procedure or as desired by the operator.

D. Provide real-time displays for evaluation and control purposes.

E. Provide for on-demand display of arbitrary information.

F. Allow on-line procedure development and modification, within judicious limits.

The specific characteristics of the control and display interface and the means of implementing it are not discussed here, as these are treated in some detail in a separate section of this report. The requirements are relevant to the checkout language, however, to the extent that provision must be made for accepting and processing the operator inputs and for generating the necessary display information.

7.3 LANGUAGE DESIGN

7.3.1 Design Objectives

Based upon the knowledge gained from the evaluation of many test languages, both existing and proposed, the functional requirements analyses summarized in Section 7.2; experience with current automatic checkout systems; and, perhaps most importantly, a projection of the manifold problems of programming on Earth into the acutely demanding environment of a Space Station, it is concluded that the ultimate effectiveness of the Space Station checkout language depends squarely on the realization of these basic attributes: easy, readable, simple, versatile and modular. The language proposed in this report is presented as a significant step in that direction. Table 7-4 expands on these basic attributes. Additional objectives are implied by elements of the operating concepts, Section 7. 3. 2.

7.3.2 Operating Concepts

It is centralizing tenent of the design that all checkout programs be constructed in the statement language. For the most part, these programs will be formulated on the ground and carefully verified, using test hardware (such as the Functional Integration Tool) or software simulations prior to use on the flight vehicle. Program modules will be named and stored, to be recalled as needed for execution, modification, or incorporation into new larger

Table 7-4

CHECKOUT LANGUAGE DESIGN OBJECTIVES

- Easy:
  - To learn (being natural, it lends itself to programmed instruction)
  - To use (via a conversational, optionally tutorial man-machine interface)
- Readable, English-like, thus providing:
  - Self-documentation
  - Comprehensibility by other users
  - Insulation against user turnover
  - Easy maintenance
- Simple as feasible, in order to:
  - Facilitate utilization
  - Reduce error potential to an arbitrary minimum
- Versatile and flexible, allowing language to be adaptable to:
  - Any Space Station problem-solving requirement
  - Cognitive styles of any potential user, such as:
    - Program development personnel on ground
    - Flight test engineers
    - Astronauts
    - Principle investigators and other scientists
- Modular and multilevel, providing compatible levels of user involvement for
  - Fast, easy selection and execution of canned routines
  - Easy composition of modified larger program units from preprogrammed modules
  - Original program development with a statement language
  - Extension of the repertoire at any of the language levels

modules. Execution of the modules may be on demand, or completely automatic, except as manual operations are provided for by design within the modules. (Examples of this would be where a manual control action is required by the automatic procedure or where crew decisions are required.) Provision will be made of course, through design of the language and the interactive control and display consoles, to maintain crew cognizance of automated procedure activity and to allow crew moderation if desired or necessary.

In addition it is foreseen that accommodation of long-term and largely autonomous operations of the Space Station will require that some procedure modifications and construction of new procedures be performed on orbit. It is also recognized that such operations are fraught with hazardous potential, and that careful controls must be designed to minimize them.

On-line operations (statement execution, procedure construction, and modification) may be performed either in the normal interactive mode or in the so-called manual mode.

In the normal mode, the software will provide tutorial assistance, guiding the operator by presenting him with option lists (menus) from which to choose valid alternatives, and providing explicit cues for operator data inputs and the like. The computer may disregard any improper option selections or data entries that it is able to detect, and provide appropriate error messages. This mode is intended for use by persons not deeply trained in programming or in the intricacies of the language.

In the manual mode, the operator enters complete commands directly via the keyboard. In this case tutorial assistance should be an option, because it could prove simply bothersome to a veteran user of this mode. If provided, the computer prompting should be limited in any case to a skeletal specification of the command, after the user has entered the operator part of the command, unless he requests at any point a complete list of the valid types of entry next to be made. For example, after entering the operator

POINT, and having allowed minimal tutorial assistance, the display might come up as follows:

INSTRUMENT AT OBJECT [TIMING] [RECORDING]
POINT

The valid designations for "object" now depend upon what "instrument" he is going to name (and may include a celestial direction), and the optional phrases for "timing" and "recording" depend very much on both "instrument" and "object," and may even not be applicable. However, an effective operator will quickly reach a stage where the above cues will be completely adequate. Sometime after that, their presence on the display will only be annoying, so he should be able to disable the prompting line.

Extension of the program repertoire at the various language levels is proposed as follows: In the normal mode, no new canned routines can be added, because the logic for reaching each of them forms a complex program that must be unconditionally protected (even though it can be a module originally written in the statement language), because the consequences of an incorrect modification could be disastrous to Space Station operations. It may be permissible, however, to allow modification of some of the canned routines, at least among those having to do with noncritical operations, so that additional functions are performed (optionally) by the routine.

In the manual mode, new modules may be formed, subject to restrictions on constituent old modules or interlocks on usage of the statement language. For example, GNC modules may not be modified or used in any combining process. However, a useful new sequence of basic functions of an experiment should possibly be allowed. The statement language is also provided with limited means for extension, as described in the specification.

7.3.3 Fundamental Characteristics

These are listed in Table 7-5 and were taken for the most part from those identified in the section of the same name in the language survey. Characteristics carried over from the former list are considered desirable as defined in the referenced section, with qualification. Those not appearing

Table 7-5

FUNDAMENTAL CHARACTERISTICS FOR SPACE STATION CHECKOUT LANGUAGE

- English-sentence-like statements
- Abbreviated language vocabulary:
    - Optional for operators
    - Required for engineering units
- English-like variable names
- Coded test point designations
    - Optional alternative to Engineering name in input
    - Option of all-names or all-codes in output
- Dictionary of instrumentation data
- Multiprogramming capability
- Monitoring (concurrent, software)
- Delays concurrent with processing
- On-line data input capability
    - By program or operator
- On-line statement input and execution
- On-line program change capability
- On-line program development capability
- Selective control of test progress
- Full variety of internal data types
- Bit and character manipulation
- Computational capabilities
- Language extensible without compiler changes
- Language partitioned into user-oriented subsets

here were deleted because of conflict with the design objectives. The characteristics appearing for the first time in the present list are discussed below.

7.3.3.1 Delays Concurrent With Processing

It is useful at certain points of a real-time process to be able to delay execution of a future step a certain length of time. This can be programmed explicitly, given that time values can be sampled, but it can probably be implemented easily as a special feature. This may utilize the assistance of the event scheduler of the executive operating system.

7.3.3.2 On-Line Program Development Capability

Beyond "on-line program change capability," this requires only the capabilities to open a new entry in the program module ledger, and to allocate or disallow space for the result. (The latter capability is, in fact, already needed for program modification where statements are being added.)

7.3.3.3 Full Variety of Internal Data Types

The crucial types implied by this characteristic are hardware floating point numbers (FORTRAN REAL's) and the table data structure, i.e., a two-dimensional array with provision for reference, not only to individual elements and to the entire structure but to a whole row or column at a time. In the time frame of the Space Station, there will be no perceptible economic or mission performance penalty in utilizing hardware floating point for mixed numbers or even for integers.

7.3.3.4 Bit and Character Manipulation

By this is meant provisions for access to, operations on, and arbitrary movement of, single or selected groups of bits or characters in a string. A corollary capability is that for input and output of such strings, as for support of telemetry.

7.3.3.5 Computational Capabilities

Some users will require some fairly sophisticated capabilities for operations on data, that would only confuse other users. It is proposed that the compiler actually dintinguish between users, whose problem-solving responsibilities will be established in the compiler data base. The basic language subset will be suited to value assignments, such as Y = A + X + B.

7.3.3.6 Language Extensible Without Compiler Changes

Some practicable ways to implement this characteristic are proposed in the language definition, namely by use of the SUBROUTINE and specify statements. Further discussion is deferred to the parts of the definition describing those statements.

7.3.3.7 Language Partitioned into User-Oriented Subsets

The requirement for this characteristic arises from the potential richness of the statement language; for, if it evolves to become adequate for handling the general DMS control and concomitant data processing, it will possess sufficient capabilities to implement any Space Station function now foreseeable.

This does not compel any user group to use the statement language, but it does submit the fact that all groups could use it, subject only to the preexistence of callable input-output routines and a compiler. The extensibility property of each language level, and the richness of the complete statement language, implies the capability to compose almost any new special-purpose function needed. It might be claimed that a systems program, such as an executive, would be too inefficient with regard to both execution time and storage space if written in this language. That possibility cannot categorically be denied, but such inefficiencies can demonstratably be sharply limited by intelligent compiler design. Computer technology of the realistic time frame of the Space Station could conceivably render inconsequential any irreducible time and space inefficiencies of compiled code.

Whether or not a universal statement language for all Space Station functions evolves, it is the working assumption of this Task that one will be used for the development of both onboard checkout and experiment control program modules. There will be wide differences between the language needs of test engineers and scientists, however, and even between the engineers responsible for different subsystems and between scientists concerned with different experiments.

Because of the very real potential for unsuccessful training and lack of acceptance should the whole statement language be presented to all personnel who need any part of it, the conclusion here reached is to design a full-capability statement language with all consistency possible, but then to partition it into logical and functional subsets, and to disclose to each user only the subset or group of subsets that are essential to the fulfillment of his specific responsibilities.

It will be seen in the language definition that the partitioning of the language must be both horizontal and vertical in order to be effective; i.e., one kind of subset consists of a specific set of statements with some or all of their variations. Another kind of subset is defined by the type of variations it would support in statements belonging in basic form to other subsets. Determining the optimum partitioning will be evolutionary, but based on experience, it will be worth the effort. A first cut at allocating statements to subsets forms part of the language specification, with a working set of subsets defined as follows:

Code	Subset Name	Relevance
GP	General Purpose	Basic data processing statements and options.
CC	Checkout and Control	Basic real-time problem-oriented statements.
ME	Multi-Element	Allows specified manipulations of multi-element variables within a single statement.
AF	Advanced Formulation	Provides for formulas containing other than just real arithmetic operations or more than one basic Boolean condition.
VM	Vector-Matrix	Provides for operations on three-component vectors and 3-by-3 matrices.

Code	Subset Name	Relevance
SM	Symbolic Math	Allows possibility of recognition and implementation of the function of special symbols, such as the integral sign.
AG	Advanced General	More sophisticated data processing facilities
AC	Advanced Checkout	More sophisticated real-time capabilities

7.3.4 Detail Characteristics

These are listed in Table 7-6 on the following pages. The characteristics covered include all of those in Table 7-2 in the language survey report, with the exception of the Time field, whose function can be adequately implemented otherwise without giving up so much space to a fixed field in each statement. The main entry, Multi-Element Arrays, is expanded with subsidiary entries introducing an enriched variety of such structures. Also, label, character and bit stream type variables are added as is the label constant, to provide the content of label variables.

7.3.5 Statement Definition

The following is a preliminary definition of a statement set that embodies the design requirements and characteristics previously derived. The proposed language also offers, as options, capabilities suited to expanded use as a general purpose task oriented language.

Table 7-7 lists the proposed statement set for the Space Station checkout language by category and operator, and shows the apportionment of prime and optional subsets of the language.

It may seem at first glance that a superfluity of statements has been proposed. It will immediately become apparent, however, that there are only a few in the general-purpose (GP) and checkout (CC) subsets, which in combination are adequate to develop a majority of checkout programs. Implementation of the other subsets would be optional to different extents. The subset SM (symbolic math) is suggested, for example, just to indicate how advanced program applications can become.

Table 7-6 (Page 1 of 6)

DETAIL CHARACTERISTICS OF SPACE STATION CHECKOUT LANGUAGE

Characteristic	Description
Fundamental Concepts	
Program structure	Segmented "modules" nestable to TBD levels; called modules are concatenated at level of caller; a nested module consists of none or more statements enclosed between BEGIN - END. A saved module can then be inserted anywhere. A function module produces a valve.
Multiprogram levels	TBD, plus TBD variable monitoring, TBD trend analyses, TBD executive tasks.
Globality of variables	Global throughout main program or within any called module, except that LOOP index variable is global only within loop.
Overlaying of segments	Automatically made just before a non-resident segment is needed. Segments are dynamically determined as a multiple of TBD words, according to available main memory. Modules may be forced resident up to the capacity of main memory.
Character Set	
Upper-case letters	Yes
Digits 0-9	Yes
FORTRAN special set	Yes
Others	Single quote (apostrophe) standard. Many others optional; e.g., lower-case letters, all other ASCII, math symbols, such as integral sign, superscripts and subscripts.

Table 7-6 (Page 2 of 6)

DETAIL CHARACTERISTICS OF SPACE STATION CHECKOUT LANGUAGE

Characteristic	Description
Statement Format	
Control characters	Single capital letter, first non-blank encountered. H, S, N, E, R, B, C, or TBD. Multiple control characters per statement permitted.
Comments	Remainder of statement if * first encountered.
Label	Alphanumeric string beginning with digit; first non-blanks or encountered after control character and at least one blank.
Operator (fixed field)	Capital letter string recognized as one of the defined operators, or its first two letters; first non-blanks, or encountered after control character and/or label and at least one blank.
Variable field	Remainder of statement after operator and at least one blank encountered.
Continuation	New line introduced by separate character as first non-blank.
Constants	
Integer (decimal)	Digit string not containing ".". May be converted to floating point number, either unconditionally or based on context.
Mixed Number (decimal)	Digit string having "." in front, back, or imbedded. Optional trailing power-of-10 multiplier. Always converted to floating point.
Octal	Octal digit string terminated by "K."
Binary	Binary digit string terminated by "B."
Hexadecimal	Hexadecimal "digit" string, tentatively terminated by "S."

Table 7-6 (Page 3 of 6)

DETAIL CHARACTERISTICS OF SPACE STATION CHECKOUT LANGUAGE

Characteristic	Description
Time	Sequence of one or more of the following: Appropriate digit string, blank, time-unit abbreviation, blank.
Textual	Any character string not containing single quotes enclosed by single quotes. If not utilized by context, this is imbedded commentary or for improvement of readability. Compiler ignores.
Logical/Boolean	Separate digit 1 or 0 or any of a number of TBD words, such as ON, OFF, TRUE, and FALSE.
Label	This already existed in contexts, such as in the GO TO statement, but here it is defined as any valid label used as initialization or update of a label variable.
Complex	Two mixed numbers separated by comma; enclose in parenthesis.
Multi-valued arrays	Appear as initialization or updating of any multielement variable, or as an explicit list, for multielement (ME) subset of statements. Data may be any type unless specified in the declaration of the variable. Separated by commas, and enclosed in parentheses if desired or ambiguous.
Variables	
Symbolic name size limit and naming conventions	String of up to TBD alphanumerics, with imbedded periods (".") optional. Lead capital letter required. Must be separated from other names, constants, and language vocabulary by at least one blank.
Internal variable types	Each must have a unique symbolic name, within globality. Default type if not declared is mixed number.

Table 7-6 (Page 4 of 6)

DETAIL CHARACTERISTICS OF SPACE STATION CHECKOUT LANGUAGE

Characteristic	Description
Integer	Fixed point. Loop indices I, J, K, L, M, N, pre-declared.
Mixed number	Floating point. Single or double precision. Real or complex.
Time	Value of time or time interval in milliseconds. One 32-bit word each will do.
Logical/Boolean	Values 1 or 0.
Digital (multidiscrete)	Up to one word of significant values 1 or 0. In an operation, such as comparison of this variable with a constant, the constant determines the number of bits compared. Analogous rules apply for the remaining internal variable types.
Bit stream (bits)	Up to the number of bits declared for the variable of significant values 1 or 0. Declared length and current beginning and ending bit/word positions should be retained or maintained.
Character	Up to one word of significant characters.
Textual/string	Up to the number of words declared for the variable of significant characters Declared length and current beginning and ending character positions should be retained/maintained.
Label	This is intended to hold a valid label-type constant, and must be declared large enough to hold the maximum-length label it might receive.
External variables	Each must have a symbolic name (in dictionary of instrumentation data) unique among all variable names, and also have a unique coded "test-point" name. Variable can be represented in one word by its dictionary reference number, actually another type of

Table 7-6 (Page 5 of 6)

DETAIL CHARACTERISTICS OF SPACE STATION CHECKOUT LANGUAGE

Characteristic	Description
	internal variable. Quantities sent to, or read from, test points must agree with internal variable source or receptacle type, if any declared, as follows: discrete: logical, analog: mixed number, multidiscrete: digital, keyboard-display data: textual, telemetry data: bit stream.
Engineering units	A TBD universally accepted set of abbreviations. Dictionary contains default engineering units for each external variable, so use in language is optional. If used, however, they override default and may thus be used to scale results, or to obtain in a different form; e. g., as a percentage of full scale instead of absolute pressure.
Multi-element variables	Must be explicitly declared. All may have subscripts, which are any valid integer expression in the computational subset in effect. Bits and characters may be accessed. Specific operations effecting all elements of one or more multi-element variables in one statement constitute the multielement (ME) subset of the language.
Array	Up to TBD dimensions. Data type of content may optionally be restricted by declaration.
Table	Two-dimensional arrays of one or more columns. The variable type of each column is subject to distinct declaration. A column that contains external variable reference numbers can be content-addressed, i.e., an external variable name used as a subscript in a reference to (that column of of) the table will return the row

Table 7-6 (Page 6 of 6)

DETAIL CHARACTERISTICS OF SPACE STATION CHECKOUT LANGUAGE

Characteristic	Description
	number of the element containing the associated reference number, if found. Any element, row, or column of a table may be accessed by appropriate subscripts, being an "*" in the second and first positions in the case of the row and column, respectively. A table column addressed in an appropriate ME subset statement will have the operation performed successively on all elements of the column. Rows may be added to or deleted from a table, and existing rows may be activated or suppressed.
Vector-matrix	These are special arrays specifically introduced to facilitate the handling of coordinate systems, such as would be required for the positioning of experiment modules. The vectors are of length three, and the matrices are 3 by 3. Ideally, the entire arrays will be referenced by name in formulas in which the operations are those of matrix or vector algebra. Subscripts up to a value of three may be used.
Displayed list	This is not a variable per se, but a sequence of variable names or constants, separated by commas and enclosed by parentheses, if desired or necessary to avoid ambiguity. May be used in multielement subset statements.
File	Bulk data, to be stored on, or retrieved from, mass storage devices.

Table 7-7 (page 1 of 3)

STATEMENT PROPOSED FOR THE SPACE STATION CHECKOUT LANGUAGE

Statements	Language Subsets							
Category/Operator	GP	CC	ME	AF	VM	SM	AG	AC
Program Structure								
IDENTIFICATION	P	O					O	O
BEGIN	P						O	
END	P							
FINIS	P							
ANNEX	P						O	
BLOCK								P
LEAVE							P	
Subprogram Structure								
SUBROUTINE	P						O	
FUNCTION							P	
Stimulation								
APPLY		P	O					O
SET		P	O					
RESET		P	O					
PULSE			O					P
ISSUE								P
TRANSMIT			O					P
ESTABLISH PROHIBIT			O					P
ESTABLISH SIMULTANEOUS			O					P
ADJUST				O				P
Timing Controls								
DELAY		P						O
WHEN		P		O				
UPON				O				P
ESTABLISH INTERVAL								P
ESTABLISH IMMEDIATE				O				P

Table 7-7 (page 2 of 3)

STATEMENT PROPOSED FOR THE SPACE STATION CHECKOUT LANGUAGE

Statements	Language Subsets							
Category/Operator	GP	CC	ME	AF	VM	SM	AG	AC
Measurement								
MEASURE		P	O					O
SAMPLE		P						O
ESTABLISH CALIBRATION								P
MONITOR		P		O				O
ESTABLISH MONITORING			O					P
VERIFY		P	O	O				O
Multiprogramming Control								
START				O				P
TERMINATE				O				P
ESTABLISH SYNC								P
ESTABLISH RESERVE			O					P
ESTABLISH PRIORITY								P
Programmable Sequence Control								
CALL	P						O	
EXIT	P							
GO TO	P						O	
IF	P		O	O				
DO							P	
QUIT							P	
Execution Pause Control								
HALT	P							
PLEASE								P
Operator Controls								
ESTABLISH ABORT		P						
ESTABLISH BACKOUT								P

Table 7-7 (page 3 of 3)

STATEMENT PROPOSED FOR THE SPACE STATION CHECKOUT LANGUAGE

Statements	Language Subsets							
Category/Operator	GP	CC	ME	AF	VM	SM	AG	AC
Operator Displays								
DISPLAY		P						O
PRINT	P						O	
FORMAT							P	
LIGHT		P						
Peripheral Input-Output								
LOG								P
INPUT							P	
OUTPUT							P	
Computational Capability								
ASSIGNMENT	P	O	O	O	O	O	O	O
Variable Definition								
VARIABLE	P		O		O		O	O
ESTABLISH ACTIVE								P
Language Extension								
SPECIFY				O			P	O
On-Line Commands								
COMPILE	P						O	
MODIFY	P						O	
ALLOCATE							P	O
SAVE	P						O	
EXECUTE	P						O	
DISPLAY		P						
CALL	P						O	
START		P						O
TERMINATE								P
GO TO							P	
ASSIGNMENT							P	

NOTE: P = Primary; O = Optional

In general, the statements introduced in advanced subsets AG and AC could be implemented by means of combinations of statements in the basic subsets, or by complicating or adding to the alternatives of the basic statements. In view of the design objectives of simplicity and readability, it is considered preferable for the total language to have a fairly extensive but meaningful operator vocabulary, consisting of the first words of statements whose content is straight forward, for the function named by the operator. Thus, both the language vocabulary and formats are easy to remember, and thus, they are simplified, in comparison with a statement set containing a more limited vocabulary, which consequently must be more ambiguous.

The language will be defined by statement formats that will contain explicitly specified elements ("literals"), such as the operator (statement name, "verb"), keywords (TO, BEFORE), accessory language vocabulary "globals" like TIME and engineering units), and special characters. All special characters are literals. Literal words are always given in capital letters exactly as they are to appear in the statement used. Other elements of statements are at the disposal of the user and are specified by names consisting of one or more lower-case words. If a name is more than one word long, the words are shown to belong together and distinct from those naming adjoining elements by being linked with hyphens; e. g. , "external-variable. "

Unfortunately, this fairly natural way of indicating compound names is not feasible for user-defined names, because the connotation of "hyphen" in the source input necessarily is "minus." The only standard special characters is ".", and that is what is to be used.

Three more conventions complete the shorthand rules of this specification format. (One common convention is obvious: the left-to-right order of elements in a statement format determines a like order of items in a statement constructed in that mold. This is worth mentioning only because more-ambitious language developments under way would allow virtually arbitrary positioning of phrases within a statement, interspersed with arbitrary words inserted to improve readability; e. g., English itself. The processing of such languages appears too complex to justify their proposal at this time.)

A. The enclosure of a vertical list of elements in braces $\{ . \; . \; . \; . \}$ indicates that a choice of exactly one of these elements must be made at that place in the statement.

B. The enclosure of a vertical list of alternatives in brackets $[. \; . \; .]$ indicates that a choice of one of the enclosed elements is optional at this position. (This list may consist of just one element.)

C. A superscript on either braces or brackets is in lieu of writing the entire alternative structure sequentially that many times. The superscript indicates the maximum number of appearances of the structure; in the case of the braces, therefore, all but one if written out would have to be enclosed in brackets as well. This notational device would be a convenience for low-valued superscripts, but its actual use in this specification is to indicate an indefinite number of allowed appearances, symbolized by using the superscript "n" in as compact a manner as possible. Note that this amounts to a TBD part of the specification, which in a practical compiler would have to be limited, by at least some alterable parameter.

Two special consequences of superscript-denoted repetition are to be noted. (1) The same alternatives exist anew at each repetition; the notation does not mean that one alternative is to be picked and inserted in the statement the prescribed number of times. (2) It makes sense for the number of alternatives in the superscripted structure to be only one, because this now is the way to indicate a string of elements of the same kind; e. g., $\{\text{digit}\}^{n}$ defines an integer at least one digit long, while $[\text{digit}]^{4}$ indicates an optional integer of up to four digits.

The format description of each statement is introduced by a line that has the statement operator name on the left, the primary language subset to which the language is allocated in the center, and applicable optional subsets on the right.

The allocation of a statement to primary and optional subsets is based on the explicit options that appear in the statement format and on implicit complexities that elements of the statement may assume, which will be pointed out in the subsequent discussion. The primary subset to which a statement belongs is determined by its simplest form; in some cases, none of the forms of the statement is sufficiently more complex to believe that these forms should not be revealed to the natural users of the primary subset. In others, one or more levels of increased complexity seem discernible, and allocation is made to one or more optional subsets accordingly. The particular subsets defined and the allocation of language statements and capabilities to them is certainly a matter for further refinement.

The statement format follows the line introducing it. The format always begins with the operator name, avoiding the nuisance of repeating "[control-character] [label]" as the first part of each statement format, but it must be remembered that these optional elements are a part of every statement. Furthermore, since it is entirely possible that a user may want to use more than one control character at a given point of a program, he may do so by inserting one or more lines before the statement line proper, which contains only a control character. No explicit indication is needed that the lines containing only control characters belong to the next statement following.

A short discussion of the functions of the statement follows its format. Herein will be described the meanings and nuances of the elements of the statement that may not be self-evident, including the reasons (where applicable) for allocating some forms of the statement to any optional subset.

The statement formats will be presented in the simplest and most concise form possible. Otherwise, it was found that the sense of a statement could easily be lost behind towering lists of alternatives and bewildering "explanations." It is indeed intended that the language be rich and that it possess almost unlimited alternatives of expression, but the point that should be made first is that the language is simple but effective for the typical user who needs only the basic checkout and control subset CC. Therefore, rigorous definitions of the following terms are deferred. Meanwhile, the intuitive definitions now given will serve quite well:

constant: Any string of characters in a statement having an explicit, displayed value; also, any list of constants.

variable: Any string of characters naming a variable quantity, function, or point whose current value or status is implicit in, but cannot be known from, the name; also, any list of variables.

expression: Any string of characters that, by using the established rules of evaluation and, the given values of any variables named, could be reduced to a meaningful value of some type; also, any list of expressions.

condition: Any expression, no matter how complex, whose residual value is TRUE or FALSE; also, any list of conditions.

list: Any sequence of language elements; it may be contained in a multi-element variable and referenced by its name, or it may be displayed explicitly.

7.3.5.1 Program Structure

This section describes statements and elements that give structure to programs of the statement language.

A. IDENTIFICATION Subset: GP Options: AG, CC, AC

ID[ENTIFICATION] information

This statement must be at the beginning of any program to be compiled. The information it will carry will include such things as the user's name, program or hardware configuration control codes, and installation-peculiar data.

B. BEGIN Subset: GP Options: AG

BE[GIN] program-name [RESIDENT]

This must be the first statement of any program module that is not a subroutine. The program name may be any symbolic name (i.e., variable name) unique among program names. If this program must be accessible in less time than it would take to get it from mass storage, it is requested that it be made main-memory RESIDENT. Any BEGIN statement is automatically an entry point (for purposes of the on-line GO TO) without flagging it with an E control character. This, however, does not mean that execution starting at any BEGIN statement can be guaranteed to be successful.

C. END Subset: GP Options: None

```
        [EN]D        [program-name]
```

For every BEGIN, there must be an END, and the optional program name on an END statement should be the same as the one on the associated BEGIN. A BEGIN, an END, and all the statements in between form a program "module." Just like expressions enclosed by pairs of left and right parentheses in formulas, modules delimited by BEGIN and END pairs may be nested in larger modules, but they may never overlap; i.e., when an END (right parenthesis) is encountered, it belongs to the closest previous unaccounted for BEGIN (left parenthesis), regardless of any program name on the END statement. The statements (symbols) in between constitute the body of the module (expression) which is named by the BEGIN.

Variables to be declared global to a module should appear in VARIABLE statements immediately following the BEGIN that starts the module. The scattering of declarations through a module makes the compiler work take longer than necessary, is disorderly, makes comprehension by other persons more difficult than it would if the declarations were collected in a standard place, and may cause loss of desired globality because of the subsequent insertion of BEGIN-END's. (This could show up as a subroutine suddenly unable to obtain its arguments.)

D. FINIS Subset: GP Options: None

```
        FI[NIS]
```

This is the required last statement before compilation can be requested. All statements following the last IDENTIFICATION statement will then be compiled.

E. ANNEX Subset: GP Options: AG

$$\text{AN[NEX]} \left\{ \begin{array}{l} \text{function-name} \\ \text{program-name} \\ \text{subroutine-name} \end{array} \right\}$$

The named program unit will be retrieved (if it exists) and inserted at this point, providing a Library service. There will be least two levels of Library, one containing all program units that have been composed and saved, and another containing only those constructed from language subsets that the present user has been made acquainted with. A user may view the contents of his library by an on-line DISPLAY command.

F. BLOCK Subset: AC Options: None

[BL]OCK

If a user feels that the automatic determination that would otherwise be made by the compiler would jeopardize timing requirements, this statement is used to "manually" determine the boundary between overlay segments, by requiring an overlay to be performed when there may not be time enough.

G. LEAVE Subset: AG Options: None

LE[AVE] [language-name]

The lines following this statement will no longer appear in the statement language until further notice. Usually they are in assembly language. The compiler will be prepared to compile, preprocess, or merely pass through all material until some tell-tale operator in the other language is encountered, or another LEAVE, which can conceivable be followed by material from yet another language, etc.

H. E Subset: AG Options: None

The "E" control character should be placed on statements where control may usefully and successfully be transferred by means of an on-line GO TO statement (see below).

I. * Subset: GP Options: None

When encountered as a control character, "*" signifies that the entire statement is commentary, which may utilize any of the available characters.

J. ' Subset: GP Options: None

Whenever the single quote or apostrophe (') is encountered, it and all characters following it through the next single quote are regarded as imbedded commentary, unless usable by the context as a text constant. Any characters, except the single quote, of course, may be used in the commentary, which may be very useful to improve readability of a statement or to provide explanatory material closer to the subject than a commentary line could.

7.3.5.2 Subprogram Structure

A. SUBROUTINE Subset: GP Options: AG

SU[BROUTINE] subroutine-name (parameter list) [RESIDENT]

Subroutines may be defined anywhere between the IDENTIFICATION and the FINIS statements. However, they disjoint modules from any surrounding statements, and might as well be required by programming standards to be defined between the last END (END of the current top-level program module) and the FINIS.

Likewise, needed existing subroutines should be brought into this area by means of ANNEX statements. (Certain subroutines may possibly be built in, as elementary mathematical functions would be.)

Subroutines are subject to call from any level of the BEGIN-END module, or from any subroutine, so the best globality level to assign variables within subroutines is the uppermost; i.e., program-wide. This also minimizes the number of parameters that must be explicitly named.

Use of the optional word RESIDENT will force the associated subroutine to be resident all of the time, available for instant call. The most important feature of subroutines in the statement language is the facility they provide for extending the language, particularly in the direction of readability:

1. A subroutine can be called simply by mentioned its name (if no arguments must be passed by means of the call); i.e., the CALL operator (see below) is not required if the subroutine name is unique from all operator names of the statement language.
2. The subroutine name can be lengthy, functionally descriptive, and structured like English.
3. In particular, the name can be distributed; i.e., mixed in with the parameters. Arguments replacing parameters in a subroutine call do not have to be enclosed by parentheses (see below); therefore, when the parameters are each isolated by parts of the subroutine name, the call comes out in plain engineering English, words intermingled with data. To avoid ambiguity, however, any two or more parameters not separated by parts of the subroutine name must be separated by commas, as must the arguments replacing them in a call of

the subroutine. Even this might come out looking reasonably good; viz, the subroutine introduced by the statement

```
SUBROUTINE PLOT ' ' (PARAM. 1) ' '
          AT (PARAM. 2, PARAM. 3)
```

could be called as follows:

```
PLOT ' ' * ' ' AT 27. 4, 89. 9
```

Also demonstrated here is the fact that ordinary quotation marks, if formed from two successive single quotes, can be used at will, because they simply delimit commentary, albeit null in this case.

Whether distributed or not, a subroutine name must be unique from all other subroutine names. It may consist of any legitimate symbolic name; i. e., beginning with a letter and continuing with alphanumerics with periods inserted where blanks might otherwide normally occur. Each element of the parameter list must also be a legitimate symbolic name, distinct from one another and from any other variable named in the subroutine. Consider another example:

```
SUBROUTINE COMPUTE. THE. RETROFIRE. BURN. FOR
  (SPEED, ORBIT, DESIRED. ORBIT, SPEED. RESULTING)
```

A subroutine may (if desired) contain nested BEGIN-END modules and functions.

B. FUNCTION   Subset: AG   Options: None

FU[NCTION] function-name (parameter list)

A function will be resident in main memory if the module containing it happens to be resident or is forced resident.

A function is like a subroutine in almost every respect except that basically it is called by mention in a formula, therefore requiring a result to be returned to that position, instead of being called by mention in a CALL statement, requiring nothing but execution of the subroutine. A subroutine may produce results or perform actions; if there are results, they are returned via parameters to which values are assigned by the subroutines, or via established variables, which by definition are global within both the subroutine and the calling module. Because of the nature of the context of function calls, they would be subject to serious ambiguities or recognition difficulties if distributed-name parenthesis-free calls were allowed; these calls, therefore, are not allowed. Consequently, a function call is typified by HYPOTENUSE (X,Y). A function call looks like a variable subscripted, and it is therefore necessary that all function names be distinct, at least from subscriptable variable names.

Two requirements that subroutines and functions have in common with each other and with BEGIN-END modules are (1) that they begin (after the defining line) with the declaration of any variables, as needed (because they are multidimensional) or desired for clarify; and (2) that they end with END statements, which optionally repeat the name of the subroutine or function.

7.3.5.3 Stimulation

The following group of statements provides a means for applying test stimulus and control signals.

A. APPLY Subset: CC Options: AC, ME

AP[PLY]$\begin{Bmatrix}\text{constant}\\\text{variable}\end{Bmatrix}$ [engineering-units] [TO external-variable]

The last phrase is indicated as optional, only because the variable that follows APPLY may be the name of a table that already contains preestablished source-destination pairs. This usage, and the possibility of using displayed or implicit lists for the variable and the external-variable, constitute capabilities that belong to the ME subset, so that the option ME above and in many other statements need not be further explained. The use of the optional engineering units can be an advanced capability, belonging to subset AC, if they are used not just to enhance readability but to actually change the interpretation of the source (elsewhere, measured) value; e.g., if the dictionary of instrumentation data indicates that the standard-source (measured) value is in ohms, it is proposed that the prefix K on the engineering units cause the source to be considered in thousands of ohms, that the words or abbreviation for percent cause the indicated fraction of available resistance to be applied, and that COUNTS or some equivalent term cause the source to be considered as the preformed command value to configure the resistor network of this fictitious example. The latter two options have obvious analogies in the case of measurements also (percent of full scale, raw data in binary), and in every place that engineering units appear in the statement formats below, the above discussion is implied although this fact is not mentioned at that point.

B. SET Subset: CC Options: ME

SE[T] external-variable [logical-constant]

This is a good place to emphasize that none of the proposed language words are sacrosanct; if the prime objective of readability for improved communication and comprehension is not best served, a proposed word may be replaced, or synonyms may be at least defined. Here it is very likely that some engineers would like to use the operator TURN ON, for example. Likewise, if the logical constant value is explicit (it may be contained in a table that also contains the external variables to be SET), words like ON, OFF, HI, and LO may be preferable to the primal values TRUE and FALSE. Definition of synonyms, by a user, is just one of the capabilities that the SPECIFY statement has for extending the statement language. (If no desired logical state is given explicitly or implicitly, TRUE is assumed.)

C. RESET Subset: CC Options: ME

RE[SET] external-variable

This statement has an assumed desired state of FALSE, for the one or all variables listed or implied; therefore, it conveniently implements functions like CLEAR ALL.

D. PULSE Setset: AC Options: ME

PU [LSE] external-variable [logical-constant]
FOR time-number [EVERY time-number]

For convenience, this packages in one statement a SET and a RESET (or vice versa), separated by a time delay. If no logical constant is given explicitly or implicitly, it is assumed TRUE. Time numbers are constants that are identifiable as having a time-type value.

This statement is not provided for the purpose of controlling a square wave generator's output, even if the optional EVERY phrase is used. The assumed Space Station checkout philosophy (nonbench checking) would have special test equipment generate the exact required square waves, which would merely be turned on or off at the statement language level. At worst, the dictionary of instrumentation data would contain control information that would be utilized at a lower, checkout executive services level. In fact, if the EVERY phrase is used, it is proposed that it indicate that the pulse is to be scheduled with the indicated periodicity, while the program proceeds until descheduled by a RESET or possibly by another optional phase of this statement: UNTIL condition.

E. ISSUE Subset: AC Options: None

IS[SUE] constant

The constant, or list of constants, is to be presented verbation to the data-bus interface unit. This statement is therefore useful for writing diagnostics at the lower level.

F. TRANSMIT Subset: AC Options: ME

```
        TR[ANSMIT]   T.B.D.
```

This statement is intended to utilize and control the available telemetry capabilities. Experience has shown that it cannot usefully be defined until the precise operational characteristics of the telemetry systems are finally established. However, the need for an interface with the language is hereby recognized.

G. ESTABLISH PROHIBIT Subset: AC Options: ME

$$\left\{\begin{array}{c}\text{ES [TABLISH]}\\ \text{FR[EE]}\end{array}\right\} \text{ PROHIBIT} \quad \text{external-variable}$$

$$\text{ELSE unconditional-statement}$$

This is the first of several statements that set or clear status conditions. Here, the ESTABLISH alternative prohibits any command from naming the external variable (or explicit or implicit list of external variables) until after a FREE PROHIBIT statement naming the same variable(s) is encountered. This would, for example, prevent some obscure path of a complex control program from tampering with an established command configuration that must remain in effect until further notice. If a prohibited naming of the inhibited variable is encountered, the unconditional statement will then be executed instead of the offending statement.

(An unconditional statement is any statement that does not contain any condition and, in particular, is not a conditional branching statement. A subroutine call is a particularly useful unconditional statement in a situation of the above kind, because it enables the execution of an arbitrarily complex analysis of the situation, yet returns control to the next statement after the one in which the problem was detected, which is probably exactly where one wants to go.)

H. ESTABLISH SIMULTANEOUS Subset: AC Options: ME

ES[TABLISH] SIMULTANEOUS label

This statement requests that all of the commands requested by the statement having the named label (or statements having the labels in an explicit or implicit list of labels) to be completed simultaneously, or as near together as possible.

I. ADJUST Subset: AC Options: AF

AD[JUST] external-variable TO condition
[OBSERVING TBD ground rules]

This statement is very well suited for such operations as slewing devices like telescopes, polarimeters, and scanners. Some possible ground rules (all or some of which should possibly be built into the dictionary of instrumentation data) include the (slewing) rate, the direction, the step size for incrementally adjusted devices, the limits of adjustment, etc. This can be the statement that implements the typical POINT command of experiments, at a higher-level usage of the man-machine interface, or by means of a more sophisticated application of the SPECIFY statement. The condition phrase may be arbitrarily complex, and therefore include not only something like an aiming requirement, but also the time when the action is to be performed.

7.3.5.4 Timing Controls

A. DELAY Subset: CC Options: AC

DE[LAY] [label] time-number

When the optional label does not appear, this statement simply delays the execution of the next statement in sequence for a period of time that is at least as great as the specified time. If a label is given, the execution of the statement having that label is scheduled as an event that may not occur before the specified time elapses. The compiler will not accept the statement unless there is another statement having the named label, if given.

B. WHEN Subset: CC Options: AF

WH[EN] time condition

This clause may either stand alone as a statement, or it may be inserted into other statements in meaningful places, perhaps limited for simplicity to a prefix. In any case, the effect of the clause is a time constraint, suspending further execution until the condition becomes TRUE. It thus serves the purpose of a separate time field, but it allows for any condition on continuance of execution, and satisfies the design objective of naturalness. However, the basic form of the time condition will be simply

"universal-time-variable = time-number"

The universal-time variables are TBD, but they would include Greenwich mean time and mission time, and are called universal because they are global within the set of all programs.

C. UPON Subset: AC Options: AF

UP[ON] condition

The use of this phrase as a prefix to a statement causes the execution of the (remainder of the) statement to be scheduled upon the condition's becoming TRUE. Meanwhile, execution continues with the next statement in sequence. In particular, the condition may be the name of an interrupt: the condition becomes TRUE when the interrupt occurs. This phrase may be imbedded in any statement where it is meaningful, but it can be seen that its peremptory significance is most apparent at the beginning of a statement. Careful consideration will have to be given to the exact conditions under which the interrupting execution can be allowed, and to what it may consist of: flag setter, counter, or subroutine.

D. ESTABLISH INTERVAL Subset: AC Options: None

ES[TABLISH] INTERVAL = time-number

This statement establishes or alters the inter-statement execution interval to be at least the specified time. This statement may be reserved for the use of executive program modules.

E. N Subset: CC Options: None

This control character is associated with a statement to indicate that it is not permissible (meaningfully) to proceed from the execution of that statement to anything but the execution of the following statement, because of timing considerations known to the user entering the control character. As a rule, execution requirements of higher priority can arise. If so, the executive should trigger a software interrupt so that an appropriately programmed UPON statement will be able to take any possible remedial action.

F. ESTABLISH IMMEDIATE Subset: AC Options: AF

ES[TABLISH] IMMEDIATE UNTIL $\left\{\begin{array}{l}\text{label}\\ \text{condition}\end{array}\right\}$

This statement is equivalent to putting an "N" control character on itself and on every following statement up to the statement having the named label, or on all statements until the condition is satisfied. The simpler use of the statement, therefore, is to provide an alternative to entering a great many N's, while the advanced use provides for nonintervention for a length of time that cannot be known in advance, and hence cannot be preprogrammed with N's. Again, if at any point the executive is unable to provide the immediate execution requested, it should provide the requesting program with that information so that it will be able to act on it.

7.3.5.5 Measurement

In the statement language mentioning the name of an external variable in any condition or expression automatically evokes the measurement of the current state or value of that variable. In addition to the means provided by the following statements for saving the values of external variables, the value of the last external variable whose name was mentioned is always available in a specific internal variable, global throughout the concerned program. The name of this variable, synonyms of which may be defined, must be the same in all programs originally, and is proposed to be VALUE or MEASUREMENT.

A. MEASURE Subset: CC Options: ME, AC

```
ME[ASURE] external-variable [INTO variable]
[engineering-units]
```

The INTO phrase is optional only if the external variable is named implicitly, by giving the subscripted name of a table instead, which table has a column identified for values. Both variables of the statement may be represented by an explicit or implicit list of variable names, thus including selected rows or all rows of a table. The optional engineering units may be significant if given, as described above. If the units are installed as a column of some table containing a column of external variable names and a column for values, the complete capability of the statement is called into action by "MEASURE table-name."

B. SAMPLE Subset: CC Options: AC

```
SAMPLE external-variable INTO multi-element-variable
    EVERY time-number    [FOR time-number]
    [ANALYZE FOR trend-measure]
```

This statement measures the value of just the one named external variable at the stated periodicity, and sequentially stores the values in the named area. This process terminates when the area is filled or when the time limit specified by the FOR phrase is reached (if the phrase was given and that time limit is reached before the area is filled). Certain trend analyses may then optionally be performed, the term "trend-measure" implying a list of one or more of a set including at least the following: average, maximum, minimum, and time gradient. Other measures of possible interest or value might be as follows: standard deviation, mode, and parameters of bimodal behavior if detected.

C. ESTABLISH CALIBRATION Subset: AC Options: None

ES[TABLISH] [NO] CALIBRATION

This statement either reestablishes or removes the normal requirement to apply calibration curves to the outputs of analog sensors. Readings taken under either condition may indicate a need to recalibrate.

D. MONITOR Subset: CC Options: AC, AF

MO[NITOR] external-variable condition EVERY time-number

ELSE subroutine-call

Once encountered, this statement causes scheduling of its execution at the stated periodicity as events independent of the program containing the statement. (In practice, a small fixed variety of periodicities is settled upon.) Once the condition fails to evaluate TRUE, however, the subroutine proceeds as a separately scheduled sequence in place of the execution of the MONITOR statement.

The term "external-variable" is not separated from the term "condition" by any English keyword because the external variable named (only one per statement) is a very integral part of the condition, and it may in some forms even appear imbedded in it; e. g., "lower-limit less-than-or-equal-to external-variable less-than-or equal-to upper limit". Engineering units may be included, and as explained earlier, they may have significance. If the condition is expressed in tolerance form, the units of percent on a tolerance are with respect to the specified nominal value, as one would expect, but the nominal value, if so desired, may be given as a percent of full scale.

Scheduling of the execution of a MONITOR statement automatically stops when the program containing the statement completes execution of the lowest-nested module containing the statement. Usually there is no longer a need for it.

E. ESTABLISH MONITORING Subset: AC Options: ME

$$\left\{\begin{array}{c}\text{ES[TABLISH]}\\ \text{FR[EE]}\end{array}\right\}\text{ MONITORING label}$$

The term "label" here refers to any means by which one or more labels can be indicated: a list of one or more explicit labels; a list of one or more variable names, each variable being one that can contain labels; or a mixed list of labels and label variables. The intention is the provision of arbitrary flexibility in turning the monitoring of external variables on and off, whole groups at a time. The referenced MONITOR statements may be anywhere in the program, and need appear only once, no matter how many times they are referred to in statements of the type being described. Each ESTABLISH or FREE variation of this statement may refer to MONITOR statements as convenient to the user; no pairing is needed unless it suits the purposes of the user.

Similarly as for the MONITOR statement, monitoring initiated by the ESTABLISH MONITORING statement ceases when execution of the lowest-nested module containing the ESTABLISH statement is completed.

F. VERIFY Subset: CC Options: AC, AF, ME

$$\text{VE[RIFY]}\quad \text{condition}\quad \left[\text{BEFORE}\ \begin{Bmatrix}\text{time-number}\\ \text{time-condition}\end{Bmatrix}\right]$$

ELSE unconditional-statement

This one statement will obtain the values of one or more variables and apply the same or individual tests to each of them, all in accordance with single or multi-element constants or variables used in the "condition" as explicitly entered. Each test can be a combination of conditions, by use of the logical operation AND, etc. If the (multiple) condition is not met, or if the BEFORE phrase is present and satisfied before the condition is met, the specified unconditional statement is executed. Otherwise, the next statement is such that it could be a whole checkout program of sorts (all in one statement), but its basic form would be very simple, for example:

VERIFY PRESSURE = 14.7 +- 10 PCT, ELSE FIXIT

7.3.5.6 Multiprogramming Control

A. START Subset: AC Options: AF

$$\text{ST[ART]}\quad \text{program-name}\ \left[\begin{Bmatrix}\text{WHEN}\\ \text{UPON}\end{Bmatrix}\ \text{condition}\right]$$

Execution of the named program is started as a separate time-share shared sequence, subject to the constraint of the optional phrase if present (which may optionally appear as a prefix).

B. TERMINATE Subset: AC Options: AF

TE[RMINATE] program-name

The execution of the named program is terminated in an orderly manner. That is, it is not allowed to run to completion, but it is allowed to run forward to the appropriate point to execute the first of one or more back-out sequences, which will restore all conditions insofar as possible to the way they were when execution of the pro-program was started.

ESTABLISH SYNC   Subset: AC   Options: None

ES[TABLISH] SYNC [POINT] integer [WITH program-name]

This statement causes the announcement of the reaching of a specific synchronization point to all programs started by this program, or to the program that started this one. Execution of the starting and all started programs is suspended until they have all reached this same synchronization point in their execution, except that if the optional WITH phrase is present, the only programs concerned are this one and the ones named in the WITH phrase. (It may contain a list of program names.)

D. ESTABLISH RESERVE   Subset: AC   Options: ME

ES[TABLISH] RESERVE [ON] external-variable
ELSE unconditional-statement

This is similar to the ESTABLISH PROHIBIT statement, except that it requests the allocation of the named external variable (or explicit or implicit list of external variables) to the exclusive use of the present program, thus inhibiting other programs from using them. If the request cannot be granted, the unconditional statement is executed.

E. ESTABLISH PRIORITY   Subset: AC   Options: None

ES[TABLISH] PRIORITY integer

This requests the executive to give the present program the specified priority in matters of scheduling. The request is always granted. However, if the number of programs operating in the higher priority levels leads to actual interstatement execution intervals greater than the TBD maxima, the executive will force a message, listing all executing programs by priority level and requesting relief in some form. Such problems should be detected during simulation or other verification activities during the program development phase on the ground. It is not anticipated that entry of the ESTABLISH PRIORITY statement will be allowed while on-orbit.

7.3.5.7 Programmable Sequence Controls

These controls are all of the data-processing type, although many of the statements primarily designed for checkout and control also provide programmable sequence control. At least those that belong to the GP subset are really necessary in the construction of meaningful, significant programs in any area of speciality.

A. CALL Subset: GP Options: AG

[CALL] subroutine-name argument-list

As already discussed under SUBROUTINE, the operator of this statement need not literally appear; and if the subroutine name was defined as being mixed in with the parameter list, it is here mixed with the arguments in the same way. The arguments may be any expressions, provided only that the data type of an argument expression after it is evaluated as far as possible agree with the data type declared in the subroutine for the corresponding parameter. (The same applies to function arguments and parameters.)

B. EXIT Subset: GP Options: None

EX[IT]

This statement is used for the logical termination of functions, subroutines, and programs. Unlike the physical END of those units, it is perfectly possible to have more than one logical EXIT. Any exit from a function must have an assignment of value of the proper data type to the name of the function immediately preceding it; the exit then carries that value back to where the function was called by name. Any exit from a subroutine causes execution to resume, with the statement following the subroutine call. Any exit from a program will cause such status to be set for that program, and a software interrupt will be triggered if the starting program has arranged for it by means of an appropriate UPON statement.

C. GO TO Subset: GP Options: AG

GO [TO] label

The statement having the named label (or the label designated by a subscripted label variable) is the first of a new sequence to be executed.

D. IF Subset: GP Options: AF, ME

IF condition [IS] $\begin{bmatrix}\text{TRUE}\\ \text{FALSE}\end{bmatrix}$ [,] [THEN] unconditional-statement

This statement causes execution of the unconditional statement if the total condition (including cases where the optional FALSE is present) is TRUE, whereas the VERIFY statement (not considering the BEFORE phrase) does the opposite. Provision of both statements in the language is not really redundant, however, because of the naturally opposite processes of data processing versus real-time control programmers, when it comes to branching instructions.

The format of the IF statement above exhibits four optional elements, none of which has any bearing on the operation of the statement except the alternative choice FALSE. However, the inclusion of any or all of the others, it might be agreed, improve the readability of the statement. It is intended that all statements will allow the inclusion of certain so-called "noise" words (such as these) to the extent that the improvement of communication and comprehension counter balances any nuisance of entry. The approved noise words must be natural and unambiguous. As seen above, they will not have to be enclosed in single quotes, as is the case for gratuitous material in general.

E. DO Subset: AG Options: None

DO $\left\{\begin{matrix}\text{THRU}\\ \text{THROUGH}\end{matrix}\right\}$ label FOR index-variable FROM integer-expression TO integer-expression [BY integer-expression]

This statement defines a program loop ending with the statement carrying the specified label. The remaining phrases are optimized for clarity, by providing each with a short keyword; and for consistency so that the increment size, which is often omitted in favor of the default value of one (1) is at the end. (In FORTRAN it is between two other values, none having keywords; when it is omitted, there is no indication except that there are only two numbers, the second of which would otherwise be the third.) Integer-expressions are any expression that evaluates to an integer value, including of course, explicit integers. An index variable is any of the automatically declared integer-type variables I, J, K, L, M, or N. There is no WHILE clause in this loop-control statement; its function is served and more clearly so by an immediately subsequent VERIFY-ELSE QUIT statement.

F. QUIT Subset: AG Options: None

QU[IT]

This statement conveniently provides for immediate termination of a loop and transfer of control to the statement following the loop, instead of having to give that statement a label, and using a GO TO. A loop may be terminated, however, by any transfer of control out of its range that does not provide for return automatically (as do subroutine and function calls).

7.3.5.8 Execution Pause Control

A. HALT Subset: GP Options: None

HA[LT] ['text']

Execution of the program containing the HALT is suspended. If the optional text is present, it will be displayed. Several on-line actions taken, as discussed in another section, may now be taken. In addition, a simple RESUME function may be activated, in which case execution proceeds with the next statement in sequence.

B. PLEASE Subset: AC Options: None

PL[EASE] ['text'] INTO variable

This statement is expected to be English-sentence-like in practice, because the text will very typically, beginning with "ENTER", followed by the description of data to be keyed in, which when done will be put into the named variable. Other possible actions, with corresponding different inputs, are toggling of a switch, depression of a function button, use of a light pen, etc. Capabilities are provided for bit and character-oriented searches of the resulting contents of the named variable, if necessary.

7.3.5.9 Operator Controls

This section assumes the existence of some operator controls that will be very useful at least during checkout on the ground.

A. ESTABLISH ABORT Subset: CC Options: None

ES[TABLISH] ABORT PROCEDURE subroutine-name

If the ABORT or equivalent button is depressed, each executing program should have established an emergency procedure to be executed by means of this statement. The compiler should refuse to compile a program that does not contain this statement at least once, before any commands are issued.

B. ESTABLISH BACKOUT Subset: AC Options: None

ESTABLISH BACKOUT PROCEDURE subroutine-name

At the beginning of each new logical section of a test program, this statement should be used to name a subroutine to be used if the operator for any reason decides to back out of that section, and indicates so by depressing the appropriate button. In order that the backout procedure have an orderly context in which to start, execution will first proceed forward in the sequence of statements currently being executed, until the first statement that has an H, S, or B control character is encountered. The back-out procedure will then be executed, and execution of the current program will be suspended at the last previous statement that had an S control character.

C. S Subset: CC Options: AC

The advanced use of this control character in checkout has just been described. Generally speaking, it is put on statements that are safe stopping points (before execution of the statements). In a more-basic use, depression of an appropriate button will cause the program to halt at the next encountered statement that carries an S control character.

D. H Subset: CC Options: AC

This advanced use of this control character was also covered under the discussion of ESTABLISH BACKOUT. Generally speaking, H control characters are placed at closer intervals than S characters, on statements that are safe for temporary holds (before execution of the statements carrying it). In a more basic use, depression of yet another button will cause the program to halt at the next statement encountered that carries an H or S control character.

E. B Subset: AC Options: None

The only use of this control character was also covered above. It is placed between H and S characters, at points where there is some degree of stability, so that a well-designed back-out procedure will be able to determine the actions to take to restore the former conditions, before present conditions can get out of hand.

F. R Subset: CC Options: None

When the program being controlled at a given operator position has its execution suspended, if the REPEAT button is depressed, execution will be initiated, starting at the last statement encountered that bears an R control character, if any. If not, a message to that effect will be displayed.

7.3.5.10 Operator Display

A. DISPLAY Subset: CC Options: AC

DI[SPLAY] [ON device-name] position-information display-information

The optional ON phrase must be used if the display device to be used is not the prime display device for the operator position controlling the program containing the DISPLAY statement. The rest of the statement needs to be the subject of an independent study in the next phase, to define a comprehensive display language.

B. PRINT Subset: GP Options: AG

PR[INT] [ON device-name] print-information spacing-information

Compared with the DISPLAY statement, this statement is relatively simple, but there are some problems, such as how to indicate in natural but concise language that there are to be six fractional digits in one conversion, that the next four numbers are to be formatted the same way, and that the next 26 characters are to be printed "as is."

C. FORMAT Subset: AG Options: None

FO[RMAT] format-information

This is included to indicate that as a fallback, the horrendously unnatural language of the FORTRAN FORMAT statement will probably find use.

D. LIGHT Subset: CC Options: None

LI[GHT] indicator-name

This statement will be needed to light passive indicators and buttons having lights.

7.3.5.11 Peripheral Input-Output

A. LOG Subset: AC Options: None

LO[G] variable

This statement will log out the named variable (or implied list of variables) along with a time stamp to the on-board recording device.

B. INPUT Subset: AG Options: None

IN[PUT] [file-name] FROM device-name INTO variable

This will be used for everything from reading the card reader (on the ground) to retrieving a named file from mass storage.

C. OUTPUT Subset: AG Options: None

OU[TPUT] [file-name] TO device-name FROM variable

This may be used for such varied purposes as storage images and driving the voice synthesizer or other audio devices.

7.3.5.12 Computational Capability

ASSIGNMENT STATEMENTS Subset: GP Options: CC, ME, AF, VM, SM, AG, AC

VARIABLE = EXPRESSION

The variable (or implicit list of variables) on the left is "assigned," or given, the values of the expression (or explicit list of expressions, or implicit list of constants) on the right. The basic use of this statement would be of the type "Y = A * X, + B." The numerous possibilities for advanced and varied use are presented in Table 7-8.

Table 7-8

ALLOCATION OF ASSIGNMENT STATEMENT TYPES AND CAPABILITIES TO LANGUAGE SUBSETS

	Assignment Statement Type					
Capability Levels/Features	Arithmetic, Double Precision	Complex	Logical	Time	Text	Mixed
Basic	GP		GP	CC		
Advanced	AF	AF	AF		AG	AC
Multi-Element	ME	ME	ME	ME	ME	ME
Bit-Byte Accessing			ME			ME
Char.-String Accessing					ME	
Content Addressing			ME	AC	ME	ME
Vector-Matrix	VM	VM		VM		
Symbolic	SM	SM	SM			

7.3.5.13 Variable Definition

A. VARIABLE Subset: GP Options: ME, VM, AG, AC

```
VA[RIABLE] [MULTI-ELEMENT-TYPE] [MODE]
    |VARIABLE-NAME  [DIMENSION]   = [CONSTANT]|
```

All variables must be defined to the compiler in some manner. However, all external variables will be automatically defined in the dictionary, and they do not have to be named in VARIABLE statements. A computer-internal variable must appear if it is of multi-element type, so that the compiler will know how much space to allocate the variable. The space is the product of numbers given

9

in the "dimension" list). Dimension and multi-element type are required for multi-element variables.

A single-element variable need not appear in a VARIABLE statement if the user can accept the default data type of floating-point, the globality mode of "local" (i.e., global only to the smallest program unit containing usage of the variable), and does not wish to initialize the variable with a constant at the time of definition. Any variable may be named in a VARIABLE statement, however, in which case the data type will be floating point if not specified, mode will be local if not specified as global, and will have an undefined value status if not initialized.

When a multi-element variable is named in a VARIABLE, the same default applies, concerning data type, mode and initialization, if not specified. However, a list of data types may be given, so that (for example) the contents of each column of a table-type variable may be defined separately. Initialization if desired may be to the same single constant value, or to a specified list of constants.

Initialization if used is restricted only to agree with the data type or types that are in effect, default or otherwise.

The subscript "n" on the part of the statement format enclosed in braces indicates that an indefinite number of variables may be named and optionally initialized in the same VARIABLE statement, except that they all have the same data type and mode, and multi-element type if applicable, which is in effect for that statement.

Very few variable names are reserved by the language, such as the names of universal time variables and the index variables, I, J, K, L, M, and N, which are automatically defined as type integer for use in DO loops.

B. ESTABLISH ACTIVE Subset: AC Options: None

ES[TABLISH] [NOT] ACTIVE TABLE-VARIABLE-NAME
(EXTERNAL-VARIABLE)
[ELSE unconditional-statement]

This statement assumes normal usage of the table variable, in which the first column is loaded with external variable names. The active and inactive state of a row of the table will be indicated by some appropriate means, such as the state of the sign/lead bit of the words of the first column, since not all of the bits will be required to hold the reference number. A content-addressing search will be performed against the first column, masking out the status bit. When a match is found, the status will be set active. If a match is not found in the entire column, the statement in the optional "ELSE" clause is executed, if present. If not, the next statement in sequence is executed after issuing an appropriate error message.

If the optional NOT appears, all the above discussion holds, except that if a match is found, the status will be set inactive.

7.3.5.14 Language Extension

It is recognized that it is highly unlikely that any statement set formulated at a given point in time will prove completely adequate over the entire lifetime of the Space Station. The Space Station program will grow and evolve, and so must the programming language. Capability is therefore provided to extend the language through use of the statement defined below:

SPECIFY Subset: AG Options: AF, AC

$$\text{SP[ECIFY]} \left\{ \text{'text'} \quad [\text{variable}] \right\}^{n} \left[= \left\{ \text{'text'} \; [\text{expression}] \right\}^{n} \right]$$

The superscript "n" implies an indefinite number of occurrences of terms of the type enclosed by the braces. In the form of this statement (not using the major optional portion), each optional occurrence of a variable when utilized will name a syntactic entity known to the compiler. For example, suppose that the ESTABLISH PRIORITY statement (7.3.1.6.5) did not exist, but that a requirement for it developed. The statement could be "built," using the form:

```
SPECIFY 'ESTABLISH' 'PRIORITY' INTEGER
```

This would, in effect, add the new statement to the vocabulary and make it available for future use.

In the full-length form of the SPECIFY statement, all that is to be to the right of the equal sign must define an existing accepted form that is recognized by the compiler. A new form, that specified on the left, is then made equivalent to that on the right. Each optional occurrence of a variable when utilized, in this case, names a dummy variable which is required only to be different from all the others named, and to be used at least once in the expression on the right-hand side. An example of using this form to simply define a synonym is shown below:

```
SPECIFY "PERFORM" = 'CALL'
```

Obviously the power, both good and bad, of this statement is large, and discretion must be used in its application. Properly used, however, it will add greatly to the versatility of the language.

7.3.5.15 On-Line Commands

The statement described in this section are entered at the interactive console in the same way as statements intended to become part of program modules. The essential distinction is that execution of the statements is immediate.

Some of the statements serve to initiate the various on-line user modes that allow the input of statements to programs, and others request execution of a limited set of functions; i.e., they are like one-statement programs. Note that words in the on-line commands do not have to be unique from those of the the normal statement language. On-line commands are interpreted by certain software logic, and normal statements are analyzed by the compiler. However, on-line commands are functionally the same as their program module counterparts.

A. COMPILE Subset: GP Options: AG

COMPILE

This requests the establishment at the entering operator position of the new program development mode. The other modes possible are program modification, and immediate execution. (In the conversational use of the interface, these three would come up in a display of alternatives to be selected from, when "On-Line" was selected from a higher-level set of alternatives.) Particularly when on-orbit, an absolutely fool-proof method of user identification must accompany the request of any of these on-line modes. (Fingerprint reading devices are already commercially available; voice recognition is in an advanced stage of development.) This is because the extent to which on-line modes are used must be very clearly limited, as a function both of mission phase and of user; e.g., if any program development or modification at the statement language level is allowed during flight, only very knowledgeable and responsible users should be permitted to make the changes, and the scope of their activities must be tightly restricted.

B. MODIFY Subset: GP Options: AG

MODIFY program-name

This requests the mode and right to modify the named existing program module. As a rule, program modification should be controlled more stringently than new program development, but the potential need for it is inescapable.

Once the MODIFY mode has been accepted, the named program will be retrieved, and administrative statements or buttons will be provided for displaying the areas of the program to be modified, "rolling" the display forward or backward, and specifying whether statements are to be added, deleted, or changed. The additions and changes will themselves be in terms of the statement language, the description of which will continue after this section on on-line commands.

C. ALLOCATE Subset: AG Options: AC

ALLOCATE number-of-additional-words-of-main-memory

The present user needs this statement only if he knows in advance that the standard allocation is insufficient. The request will be processed in light of comparative priorities and responsibilities of users having main-memory residency requirements.

D. SAVE Subset: GP Options: AG

SAVE program-name

After a user has compiled or modified a program (module), this statement is used to store the new program (version). The program name becomes the name of the new program, and it must be different from any existing (program) module name, or the system must request high-level authorization to delete the existing module first. A statement will be provided for that purpose, as well as for purging of files. Of course, the response may be to submit a SAVE statement with a new name. On-orbit, use of this statement will be restricted to at least those qualified to use subset AG.

E. EXECUTE Subset: GP Options: AG

EXECUTE

This statement requests the establishment of the most critical of the on-line user modes, that in which subsequent commands, if accepted, result in immediate execution of a limited set of functions similar or identical to those of the statement language. A description of these typical functions follows:

(The COMPILE and MODIFY modes are potentially more dangerous, although their results should not be available in the EXECUTE mode until approved by all members of some duly recognized review board, at least one member of which is on the ground. The danger referred to lies in the fact that a modified program, or a new program replacing an old, will sooner or later be executed automatically, assuming that there was a need for the program module in the first place. Granted that some form of certification must precede the acceptance of such a program into the system, it is difficult to imagine anything but the most trivial program development on-orbit that could be certified satisfactorily on-board.)

F. DISPLAY Subset: CC Options: None

DISPLAY display-name-code

The purpose of this on-line DISPLAY statement is only to request the initiation of predefined displays, such as those of subsystem status. The display request may be queued, based on its priority relative to the current usage of the display devices. This statement is apparently absolutely safe for execution on-orbit, and it may be entered without first establishing the EXECUTE mode.

G. CALL Subset: GP Options: AG

CALL subroutine-name (argument-list)

This statement requests the execution of the named subroutine, independent of any active or inactive program. The name and parameter list will be checked for validity as much as possible. The system should provide for the display of results (named by none or more of the arguments given), at least to the extent to which the sub-subroutines successfully complete execution.

H. START Subset: CC Options: AC

[WHEN time-global = time-number] START program-name
[EVERY time-number [FOR time-number]]

This statement requests the execution of a standard program, with optional phrases to specify the starting time, the periodicity, and the duration of periodicity.

I. TERMINATE Subset: AC Options: None

TERMINATE program-name

This statement requests orderly termination of the execution of the named program; i.e., it directs it to reset conditions to where they were initially.

J. GO TO Subset: AG Options: None

GO TO label

Entry of this statement at any particular operator position is valid only if that position has dedicated control over an active; i.e., STARTed, program whose execution is currently suspended. The statement requests the resumption of execution at the statement with the named label. The label must not only exist in the program in question, but the associated statement of the program must have an "E" control character (see below) defining that statement as an acceptable entry point. In any case, the operator takes an enormous responsibility upon himself when using this on-line command, believing that the program can continue successfully from some point other than where it is halted.

K. ASSIGNMENT Subset: AG Options: None

VARIABLE = EXPRESSION

Assignment statements in this context may be needed to alter a status, inserting additional or updated data, setting conditions to permit use of the on-line GO TO (above), etc. Once again, this represents a class of commands that need to be available, but which would be used on-orbit only with the greatest caution and in the context of maximum programmed interlocks against improper use; it is not the same as responding to a preprogrammed request for specific data, all acceptable values of which cannot of themselves lead to disaster. On the other hand, commands such as these have been proven to facilitate ground checkout of software and hardware enormously.

Section 8
IMS BREADBOARD TEST REQUIREMENTS

The IMS Breadboard is made up of the various hardware elements being developed and delivered by the contractor under the Space Station Study Special Emphasis Task, plus additional equipment to be provided from other sources. In accordance with current planning, this equipment is to become a part of the Space Station subsystem breadboard (SSB) at Marshall Space Flight Center.

The following paragraphs define the objectives and the hardware and software requirements associated with integration and testing of the IMS breadboard at MSFC. Primary emphasis is placed on the requirements associated with the breadboard elements that are products of the special emphasis task. These include the data bus, the control and display console, and the K_u-band communications equipment

8.1 OBJECTIVES

The basic objectives of the breadboard integration and testing program are to investigate and evaluate system-design concepts applicable to manned space flight and to provide a technology base for future development of flight hardware for the Space Station, RAM's, and related programs. More specifically, the objects are:

A. To verify and evaluate the design and operation of the data bus system, including the bus controller, the Data Bus Terminals, and the Remote Data Acquisition Units.

B. To develop and verify the interfaces between the data bus system and user systems. These user systems include the experiment programs and the Space Station subsystems, such as display and control, communications, guidance, navigation and control, electrical power, propulsion, and environment control and life support.

C. To develop and evaluate control and display concepts, including the application of existing and advanced control and display hardware to achieve optimized man-machine interaction.

D. To develop and evaluate data-processing concepts, including multiprocessor computer systems, data storage and retrieval methods, and other related aspects of data processing.

E. To develop the computer software necessary to implement the integrated data management concept, including data bus control, display generation, procedure execution, and overall executive control.

8.2 INTEGRATION AND TEST PLAN

The integration and test plan described here is designed to achieve a fully operational breadboard system in a series of logical steps, with each step building upon capabilities developed in the previous step. Maximum use is made of self-test and diagnostic capabilities inherent in the breadboard equipment to minimize requirements for complex special test equipment.

The test plan is divided into two parts, covering the initial installation and test of the breadboard, and the extended development tests. These are discussed in the following sections. Primary emphasis is given to the initial test phase, which is directly concerned with the hardware developed under the Space Station contract.

8.2.1 Initial Installation and Test

8.2.1.1 Objectives

The objective of this phase is to achieve an initial operating capability for the IMS breadboard hardware and software. Specific objectives are to achieve the following:

A. Demonstrate operation of the data bus under computer control. This will include two-way transfers of data between the computer and the bus, and computer controlled terminal-to-terminal data transfers.

B. Verify the capability of the Remote Data Acquisition Units to output commands and to acquire data under computer control.

C. Verify the capability of the display and control console operator to communicate with the computer via the data bus, and of the console display equipment to operate under computer control.

D. Demonstrate modulation of the K_u-band Exciter by computer generated data.

E. Develop and validate the basic computer programs required to implement the above operation.

8.2.1.2 Hardware Requirements

The hardware required consists of the following items:

A. Support computer (XDS 930)

B. Bus Interface Unit

C. Data bus modems (2)

D. Data Bus Terminals (2)

E. Data bus cables and line couplers

F. Remote Data Acquisition Units (2)

G. Display and control console

H. K_u-band RF exciter

I. K_u-band RF power amplifier

J. Ancilliary equipment

A block diagram of the breadboard is shown in Figure 8-1.

The data bus equipment, RDAU's, display and control console, and RF equipment will be delivered by MDAC as part of the IMS special emphasis task. The support computer and the Bus Interface Unit are to be provided by MSFC.

Support Computer

The support computer has been identified as an XDS 930. This is a 24-bit-word-length machine with a 1.75 μsec cycle time and a 32,968 word memory. Available peripherals include a typewriter, a high-speed printer, a card reader/punch, tape drives, and disk files. The system will serve as the central data processor for the IMS breadboard during the initial test and integrated phase, or until the SUMC onboard processor is available.

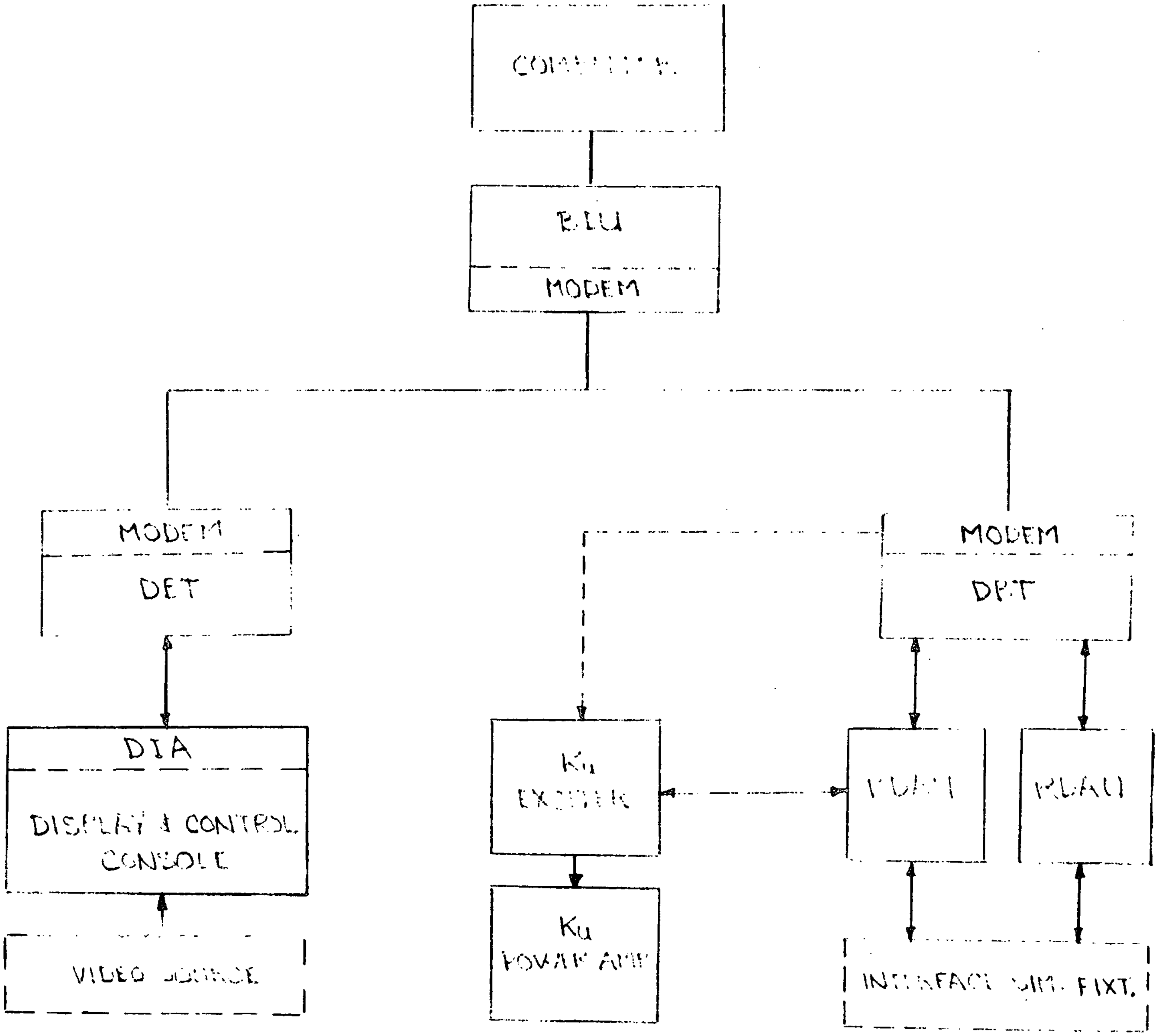


Figure 8-1. Breadboard System Diagram

Functions to be performed by the computer include data bus control, display generation, and procedure execution. The computer interfaces with the data bus through the Bus Interface Unit.

Bus Interface Unit

The Bus Interface Unit (BIU) provides the control functions necessary to interface the data bus to the computer. The unit operates under computer control, but it has the capability of independently executing computer-generated instruction sequences and buffering data into and out of the computer's main memory. The BIU also performs parallel-to-serial and serial-to-parallel conversion, timing, and traffic control functions for the bus system. The unit interfaces with the bus through a data bus modem.

Data Bus Modem

The modem performs the modulation and demodulation of data to and from the data bus. The unit receives serial digital data at rates up to 10 megabits from the Bus Interface Unit and Data Bus Terminals, and outputs the data to the bus on an amplitude-modulated carrier. It also receives amplitude modulated data from the bus and provides a demodulated serial digital data output to the Bus Interface Unit and terminals. Modems for the initial IMS breadboard will operate at a center frequency of 140 MHz. Additional digital data modems and analog modems will be added in later phases of breadboard development.

Data Bus Terminal

The Data Bus Terminal (DBT) provides a two-way interface between the bus and the using equipment(such as the Remote Data Acquisition Unit, the display and control console, and the RF communications equipment). Each DBT has the capability to accommodate eight such devices. The data interface between the DBT and the using devices is in the form of a serial digital data stream at a rate of one megabit. The terminal contains a 512-bit memory for data buffering, and the necessary logic to perform the data routing and control functions. A full description of the terminal is provided in Section 3.4 of this document.

Data Bus Cables and Line Couplers

Cables and line couplers provide the distribution of data bus signals. The cables are coaxial type. Branching of the cables is accomplished via the line couplers, which provide impedance matching and electrical isolation.

Remote Data Acquisition Units

The Remote Data Acquisition Units (RDAU) provide a standard measurement and control interface for subsystems and experiments. Capability is provided for multiplexing, signal conditioning, and digitizing of 30 analog channels and 16 bilevel channels. A built-in limit-checking capability is provided for all channels using internally stored limit values. Any or all measurement values are addressable by the computer. Sixteen bilevel command outputs are provided for control functions. The RDAU interfaces with the data bus through a Data Bus Terminal. A complete description of the RDAU is given in Section 3.4 of this document.

Display and Control Console

The operator interface is provided by the display and control console. This unit contains a variety of display devices, including both alphanumeric and graphic CRTs, LED clock/timer displays, indicator lights, analog meters, and a voice message generator. Operator input devices include alphanumeric keyboards, a light pen, discrete function switches, and a unique programmable function keyboard. The unit interfaces with the data bus through a Data Bus Terminal. It also has provision for accepting and displaying video information. A complete description of the console is given in Section 4.3 of this document.

K_uBand RF Exciter

The K_u-band exciter provides capability of accepting video, audio, and digital inputs from the data bus or other sources, and of outputing a modulated RF carrier at a center frequency of 14.5 GHz. A detailed description of the unit is provided in Section 5.1 of this document.

K_u-Band RF Power Amplifier

The RF power amplifier provides amplification of the K_u-band exciter output to a minimum RF power level of 10 watts. A full description of the unit is contained in Section 5.2 of this document.

Ancilliary Equipment

The following sections define items of ancilliary equipment that are required to implement the Phase I test and integration plan. These items include test fixtures, support equipment, and laboratory test equipment.

Interface Simulation Fixture—One test fixture, identified as the interface simulation fixture, is required. The purpose of this device is to simulate subsystem interfaces to the Remote Data Acquisition Units. These interfaces include the analog and bilevel signal inputs and the bilevel command outputs.

The analog signal inputs to the RDAU's may be in either the 0- to 40-mvdc range or the 0- to 5-vdc range. Test inputs should be in the form of variable voltage levels covering these two ranges. Although it is not necessary to provide simultaneous inputs to all 30 analog channels of each RDAU, a representative number of channels should be provided to allow a reasonable degree of fidelity in the testing. Each simulated channel input should be manually adjustable, and provision should be made for verifying the selected level by measurement with a standard voltmeter.

The bilevel inputs to the RDAU's are nominally either 0 (open circuit or ground) or 5 vdc. For test purposes, these inputs may be provided by the interface simulation fixture in the form of manual switch controlled circuits. An alternate and more economical approach would be to jumper these signals from the RDAU bilevel command outputs, thus allowing the simultaneous verification of both the input and command functions.

Testing of the bilevel command outputs will require jumpering as described in the preceding paragraph, or some form of bilevel indicators in the interface simulation fixture. These outputs are nominally 5 vdc in the "on" state and are designed to drive a 1000-ohm load. The use of lamp indicators, if desired, will therefore require that lamp driver circuits be provided in the test fixture.

Support Equipment—The Saturn DSV4B Model 298 display system has been identified as support equipment for the IMS breadboard. It is understood that this equipment will be available at MSFC and will be used to provide video inputs to the control and display console to aid in demonstration and evaluation of Space Station display concepts. The model 298 contains a film system capable of storing and selectively retrieving up to 96 35-mm slides. Slide selection is controlled by a manual control panel, which is part of the unit. The film system operates in conjunction with a flying spot scanner, which converts the film images into video form. The video signal is routed to the control and display console via coaxial cable. A video sync signal from the display and control console is also required, and this too will be transmitted by coaxial cable.

Laboratory Test Equipment—The following items of laboratory test equipment are required:

A. Oscilloscope; 50-MHz, dual trace, delayed sweep
B. Multimeter; ac/dc volts, ohms
C. Spectrum analyzer; H. P. 8551B or equivalent, with K_u-band adapters
D. RF power meter; H. P. 327A or equivalent, with thermocouple mount

8.2.1.3 Software Requirements

Software required for the initial test and integration phase will be designed to operate on the XDS 930 computer (except as noted below). It consists of the following major elements:

A. Utility routine
B. Console self-test routine
C. In-Line test and diagnostic routines
D. Executive scheduler
E. Procedure implementation routines
F. Data bus input/output control module

G. Display processor

H. Fault isolation and recovery module

A brief description of each element follows.

Utility Routines

This package consists of the XDS MONARCH and FORTLINK programs. MONARCH provides such capabilities as a batch processing monitor, peripheral equipment I/O routines, loaders, dumps, trace, and debug routines, and basic operator controls. A METASYMBOL assembler and FORTRAN II compiler are also provided. FORTLINK provides, through an off-line program called SYSGEN, the ability to generate a system on RAD or tape consisting of MONARCH and arbitrary segments of the user's code that may be rapidly called into core as needed.

Console Self-Test Routines

The display and control console includes an IMLAC PDS-1 alphanumeric and graphic CRT display. This device contains a small programmable data processor, which controls the generation of display information and has general-purpose data processing capability, including 8192 words of memory. A special test mode is provided in the display and control console logic to permit PDS-1 output data (generated under PDS-1 program control) to be transferred to the other console display devices.

Utilization of this feature requires that the appropriate test program be written for the PDS-1 processor. This program should be written in the PDS-1 assembly code, and should provide the capability to output preformatted test messages to the various display devices. The program is specifically required to perform the following:

A. Output a test message to the alphanumeric CRT display. The test message should contain all displayable characters and include all addressable screen positions.

B. Output a sequence of test messages to the voice message unit. The sequence should include each word in the voice unit vocabulary.

NOTE: Voice messages may, if desired, be simultaneously displayed on the CRT display to allow ease of verification by the console operator.

C. Output a sequence of test messages to the programmable keyboard display. The sequence should be designed to display every displayable character in each addressable screen location.

D. Output a time preset message to the event timer and exercise the timer controls.

E. Exercise the clock controls.

F. Light each of the discrete indicator lamps and illuminated switch lamps.

G. Exercise each of the analog meters through zero to full scale.

Test execution should be under the control of the console operator, either from the PDS-1 keyboard or from the PDS-1 processor maintenance panel. Capability should be provided to execute selected test subroutines as desired by the operator, to "single-step" through a test, and to repetitively loop on a given test.

In-Line Test and Diagnostic Routines

The in-line test and diagnostic routine is required to provide an early operating capability for the IMS breadboard, and for use throughout operation of the breadboard in testing and trouble-shooting the system. The diagnostics should be fully self-sufficient to allow operation independent of other software, such as the data bus input/output control and the display processing routines. The necessary capabilities are as follows:

A. Output test messages to the Bus Interface Unit and verify proper response.

B. Output all possible Data Bus Terminal commands and check the response of the terminals to these commands.

C. Load the Data Bus Terminal buffer memory with predetermined data patterns and read the data back for verification.

D. Initiate and verify terminal-to-terminal transfers on the data bus.

E. Load and verify the Remote Data Acquisition Unit memory.

F. Read each RDAU data input channel and verify operation.

G. Exercise each RDAU output channel.

H. Recognize and verify switch and keyboard inputs from the display and control console.

I. Exercise the various display and control console display devices by outputting predetermined test patterns for operator verification.

J. Output test data patterns to the K_u-band exciter.

The ability to selectively execute the various tests, to single-step through a test, and to repetitively loop on a given test should be included to provide maximum flexibility and utility. Operator control of the test sequences should be provided either from one of the display and control console devices (such as the alphanumeric CRT) or from the SDS 930 console.

Executive Scheduler

The executive scheduler, as the name implies, performs those functions necessary for overall control of the IMS breadboard test and operating programs. The scheduler receives inputs in the form of program statements from the test or operating program being exercised, and supervises statement execution. The program statements may be in high-level English-like format or in a more basic verb-noun form, depending on the degree of sophistication desired. The scheduler will perform a first-level interpretation of the program statements and then call the appropriate procedure implementation routines or I/O routines as required. The scheduler will also perform such functions as priority allocation and control, interrupt processing, and such "housekeeping" functions as error checking.

Procedure Implementation Routines

The procedure implementation routines are the "workers" of the software system, performing the various procedure tasks under the direction of the executive scheduler. These tasks include:

A. The generation of data bus outputs (commands and data) for processing by the bus control software.

B. Interpretation and processing of data received from the data bus.

C. Utilization of the display processing software to output data to the visual displays.

D. Storage and retrieval.

Data Bus Control Module

The data bus control software module provides the program linkage to the data bus, and performs the bus control functions. The module receives requests for data transfers from the procedure implementation and display processing routines. The module converts these to the proper bus formats, appends the necessary function codes and address tags, and activates the hardware Bus Interface Unit. When the transfer has been completed, the module notifies the requesting program element. The bus control module also implements the functions of bus priority control, scheduling of periodic operations such as polling, and error checking.

Display Processor

The display processor has three major functions:

A. Formatting of data to be displayed on multiformat display devices, such as CRT and plasma panel displays in accordance with the operating concepts for use of these devices. This includes conversion of data to engineering units, so labeled if required by the procedure, and the generation of graphic display formats, including coordinate conversion and curve fitting.

B. Control of discrete display functions, such as indicator lights and meters.

C. Processing of operator inputs, including alphanumeric messages, function key and discrete switch inputs, and light pen operations.

Fault Isolation and Recovery Module

This module is primarily concerned with the detection of errors in the test or operational program code, and with system errors other than those covered by the standard XDS software. Detection of a fault may simply result in notification of the operator, followed by a halt, or it may initiate procedures for isolating the fault or circumventing it so that operations may proceed.

8.2.1.4 Facility Requirements

Facility power and environmental requirements for the initial IMS Breadboard are given in the following sections. Requirements for the XDS 930 computer and the Bus Interface Unit are not included.

Environment

The breadboard equipment is designed to operate in an ambient temperature range of 50° to 85° F with relative humidity between 10 and 80 percent.

Electrical Power

Electrical power requirements for the breadboard equipment are listed below. All power is 115-vac, 60-Hz, single-phase unless otherwise noted.

A.	Data bus terminal (each)	150 watts
B.	Modem (each)	included in DBT
C.	RDAU (each)	50 watts
D.	Control and display console	1000 watts
E.	K_u-band exciter	200 watts
F.	K_u-band power amplifier	250 watts

It should be noted that the DSV4B-298, which has been identified as support equipment for the IMS breadboard, requires three-phase 115/208-volt, 400-Hz power for the logic section, plus single phase 115-volt, 60-Hz power for the video equipment. In the event that 400-Hz power is not available, the logic section may be powered by standard laboratory type dc supplies operating on 60-Hz power. The specific power requirements for this mode of operation have not been determined, but they will be slight because only a small portion of the logic section is required in the planned breadboard application.

8.2.1.5 Test and Integration Plan

The test and integration plan described in the following paragraphs is designed to achieve full system level operation of the IMS breadboard in a progression of subelement tests. This will allow verification of each hardware interface and the associated computer software in a logical sequence, and will minimize possible problems caused by introduction of multiple unknowns in a single test. In general, testing will begin at the computer/data-bus interface and progress outward to the more remote system elements. It is assumed that all individual hardware elements, such as the Data Bus Terminals and RDAU's, will have been previously functionally tested at the assembly level, and that initial off-line debugging of the software has been accomplished. It is further assumed that testing will be performed at MSFC, and that the MSFC XDS 930 computer and bus interface unit will be available in accordance with the proposed test schedules.

Computer/Bus Interface Unit Integration

The computer/Bus Interface Unit tests will verify the compatibility and functional performance of the interface between the XDS 930 computer and the BIU. Specifically, this will include the following:

A. Verify the ability of the computer to transfer control instructions to the BIU via the parallel output (POT) lines.

B. Verify the ability of the BIU to transfer data to and from the XDS 930 memory via the DSC-II channel.

Assuming that adequate self-test provisions are implemented in the BIU, the test may be performed entirely under control of the XDS 930 computer. This may be accomplished by commanding the BIU, via the POT channel, to read preformatted test data from the computer memory via the DSC-II channel and then write the data back into the computer memory at a different core location. A simple comparison check may then be performed by the computer to verify correct operation. The software for generation and verification of the test data is included in the test and diagnostic routines described previously.

Bus Interface Unit/Terminal Integration

Following initial verification of the computer/BIU interface, the Data Bus Terminals (DBT's) will be connected (one at a time) to the BIU. Under control of the XDS 930 computer, each of the DBT's will be exercised to demonstrate the compatibility of the BIU/DBT interface and to confirm proper operation of the terminals and modems. This will be the first test of the data bus under computer control, and will therefore be a critical one. The test will first establish the capability to communicate over the bus by transmitting appropriate trial command sequences to the DBT's and verifying the "A" word and "C" word responses. The DBT buffer memories and the block transfer capabilities of the system will then be verified by loading the buffers with computer-generated data blocks of varying lengths up to 512 bits, and then reading these data back into the computer via the bus, for comparison with the original data. The data transfers will be up to the maximum rate permitted by the selected BIU/computer interface configuration, but will not exceed 10 megabits/second. The software for implementing the test will be part of the test and diagnostic routines.

Terminal to Terminal Transfers

After integration of both DBT's onto the bus, the capability to execute terminal-to-terminal transfers will be proven. This may be accomplished by loading the buffer memory of one terminal with data from the computer, initiating a transfer of the data from that terminal to the second terminal, and then reading the data back to the computer from the second terminal for verification. The required software, as in the previous test, will be provided by the test and diagnostic routines.

RDAU/DBT Integration

The Remote Data Acquisition Units will be incorporated into the system, one at a time, by connecting them to one of the DBT's and executing a series of tests under control of the XDS 930 computer. These tests will verify the capability of the RDAU's to operate with the data bus and will include the following operations:

A. Load the RDAU memory with preformatted blocks of data from the computer, and read this data back to the computer for verification.

B. Read each of the analog and bilevel input channels. This will require that simulated data inputs be provided. For the analog channels, variable voltage signals (0 to 40 mvdc and 0 to 5 vdc) will be required. For the bilevels, simulated inputs (5 vdc) may be used, or the outputs of the bilevel command channels may be connected back to the bilevel measurement channels, either on the same RDAU or on the second unit.

C. Energize each of the bilevel command channels. Proper operation may be verified by connecting external equipment, such as a voltmeter, to each channel, or by connecting the command channels to the bilevel measurement channels as described above.

D. Exercise the limit-check capability of the RDAU's by varying the simulated analog inputs above and below preset limits. Verify the response of the RDAU in detecting the condition and of the DBT in recognizing the resulting "exception" signal from the RDAU. Also, verify the ability to read the RDAU out-of-limit status register from the computer.

Operation of each of the eight channels on each DBT should be checked during this test by sequentially connecting the RDAU's to all channels.

The software required to implement these operations is included in the test and diagnostic routines.

Display and Control Console Self-Test

The display and control console will have received a full functional test prior to delivery. The predelivery test will be somewhat limited in scope however, because of the unavailability of software. To reverify console operation and perform a more comprehensive demonstration of the console displays, postdelivery tests may be performed, using the console self-test routines described in Section 8.2.1.3. These tests will utilize the programmable data processor of the IMLAC PDS-1 display, which is a subassembly of the console, to generate data for exercising all of the console display devices. Operation of the alphanumeric CRT, voice message unit, programmable keyboard, clock, event timer, discrete indicator lights, and analog meters, as well as the PDS-1 itself, may thus be demonstrated prior to interfacing the console with the data bus. Since these tests are independent of the data bus and the XDS 930 computer, they may be performed in parallel with the tests of the other breadboard elements.

In addition, the video display capabilities of the console may be demonstrated during this time period by supplying an external video signal for presentation on the console monitor. This is a standard 525-line monitor and is used to display the output of the alphanumeric character generator. The external video signal may be displayed in lieu of the character generator output, or the two displays may be combined, with the alphanumeric character information superimposed over the video picture. External video may be supplied from any source capable of supplying noncomposite video at the 525-line rate and accepting separate sync from the sync generator in the display and control console.

The system block diagram illustrates use of the Saturn DSV4B Model 298 display system as the video source. This unit contains a flying spot scanner (35-mm slide system, which may be used to provide a variety of video images). Slides may be prepared, for example, to simulate experiment video, closed-circuit TV, microfilm files, or other such sources. The Model 298 also contains a vidicon camera, which may be used to provide image inputs to the console to simulate closed circuit TV or other video sources.

Console/DBT Integration

Integration of the display and control console with the Data Bus Terminal will be accomplished following the console self-tests described previously. The console will be connected to one of the DBT interface channels, and a series of tests will be performed. These tests will verify the capability for two-way communication between the console and the computer, and demonstrate operation of the console control and display devices under computer control. The following operations will be included:

A. Transmit preformatted alphanumeric messages for display on the alphanumeric CRT.

B. Transmit keyboard input messages from the alphanumeric CRT to the computer. Verify by computer comparison to predetermined formats, by output on the line printer or typewriter, for operator verification, or by return to the CRT for display.

C. Transmit preformatted alphanumeric messages and graphic forms by display on the graphic CRT.

D. Transmit keyboard input messages from the graphic CRT to the computer and verify as described in Item B above.

E. Demonstrate light pen operation by computer recognition and processing of light pen interrupts.

F. Demonstrate operation of the programmable function keyboard by computer recognition of keyboard inputs and display of preformatted computer output messages.

G. Output computer-generated messages on the voice message unit.

H. Demonstrate computer recognition of console discrete switch inputs.

I. Transmit computer-generated messages to activate the discrete indicator lights and analog meters.

J. Demonstrate computer control of the clock and event timer displays.

The computer programs associated with these operations are part of the test and diagnostic software.

K_uBand Exciter and Power Amplifier Integration

The extent to which the K_u-band exciter and power amplifier can be utilized in this phase depends to a large degree upon the availability of supporting hardware. Capability will be provided for connecting the exciter to the bus through a modem as indicated in the block diagram of Figure 8-1. Assuming that a third modem is unavailable, this will require disconnecting the data line between one of the two supplied modems and its associated DBT, and routing this line to the exciter. The control and power connections to the modem will remain connected to the DBT, since the exciter is not capable of supplying the required power to the modem. The exciter and power amplifier may thus be supplied with computer generated digital data from the data bus, and may transmit this data in the K_u band. Without a compatible receiver, however, the utility of this operation is somewhat limited. Some benefit may be gained by connecting the exciter control inputs and monitor outputs to a RDAU and verifying that these functions operate properly, and by monitoring the RF output on a power meter or a spectrum analyzer. Beyond that, a receiver of some sort will be required.

This would ideally be at K_u-band, of course, but a lower-frequency receiver could be used if a suitable down-converter were available.

For purposes of this test plan, it will be assumed that such equipment is not available, and that integration of the K_u band equipment will be limited to the operations described previously, which may be performed without the receiver. The software required to generate the required control inputs and to measure the outputs of the exciter via the RDAU will be included in the test and diagnostic routines.

Operational Software Integration

The operational (as opposed to test and diagnostic) software for the XDS 930 consists of the following elements:

A. Executive scheduler

B. Procedure implementation routines

C. Data bus input/output module

D. Display processor

E. Fault isolation and recovery module

These elements are described in Section 8.2.1.3.

Following verification of the breadboard hardware by the tests described in the preceding sections, the operational software may be integrated into the system and debugged. This will be accomplished by sequencing the software through its various subroutines and observing system operation. Both may be normal and conditional program paths will be exercised. This will require that actual or simulated system errors be induced, either through manipulation of the hardware (i.e., disconnecting a cable or turning off power on a unit) or by software simulation.

The test sequence will begin with the executive scheduler and the bus input/output module, which are essential to nearly all breadboard operations. The basic operating capability of the bus I/O module may be verified by processing test messages designed to exercise both the periodic (scheduled) and aperiodic (unscheduled) modes of operation. Linkages between the executive scheduler and the I/O module may also be verified at this time.

The display processor will then be brought into operation and verified by formatting test display messages for transfer to the control and display console. Operator-generated messages, including switch and keyboard inputs, will be transferred from the display and control console to the computer, and proper processing by the display processor will be verified. Procedure implementation routines will be tested by inputting trial program statements, verifying that the statements are correctly interpreted and executed. Testing of the fault isolation and recovery module may be accomplished concurrently with the other software tests, utilizing the module's fault detection capability to locate errors in the other software.

8.2.1.6 Schedule

The schedule for the Phase I test and integration activity is shown in Figure 8-2. The schedule is predicated upon delivery of the special emphasis task breadboard hardware in June 1972. It is also assumed that the MSFC-supplied XDS-930 computer, bus interface unit, and required software will be available at that time.

8.2.2 Extended Development Test

Following completion of the installation tests described previously, the initial IMS breadboard may be considered operational. Further testing and development may then be begun to extend the scope and capabilities of the system and, if desired, to integrate the IMS breadboard into expanded development activities, such as the proposed subsystem breadboard and the concept verification test program. The following sections discuss some of the proposed extensions of the system and the associated requirements relative to the IMS breadboard.

8.2.2.1 Computer

To develop and evaluate flight configuration computers, including multiprocessor systems, it is planned that the MSFC SUMC processor will be integrated into the IMS breadboard. The SUMC will be utilized initially as in a simplex configuration, and will later be converted to a multiprocessor configuration as additional SUMC units become available. Upon integration of the SUMC, the XDS 930 computer will assume a support and back-up role.

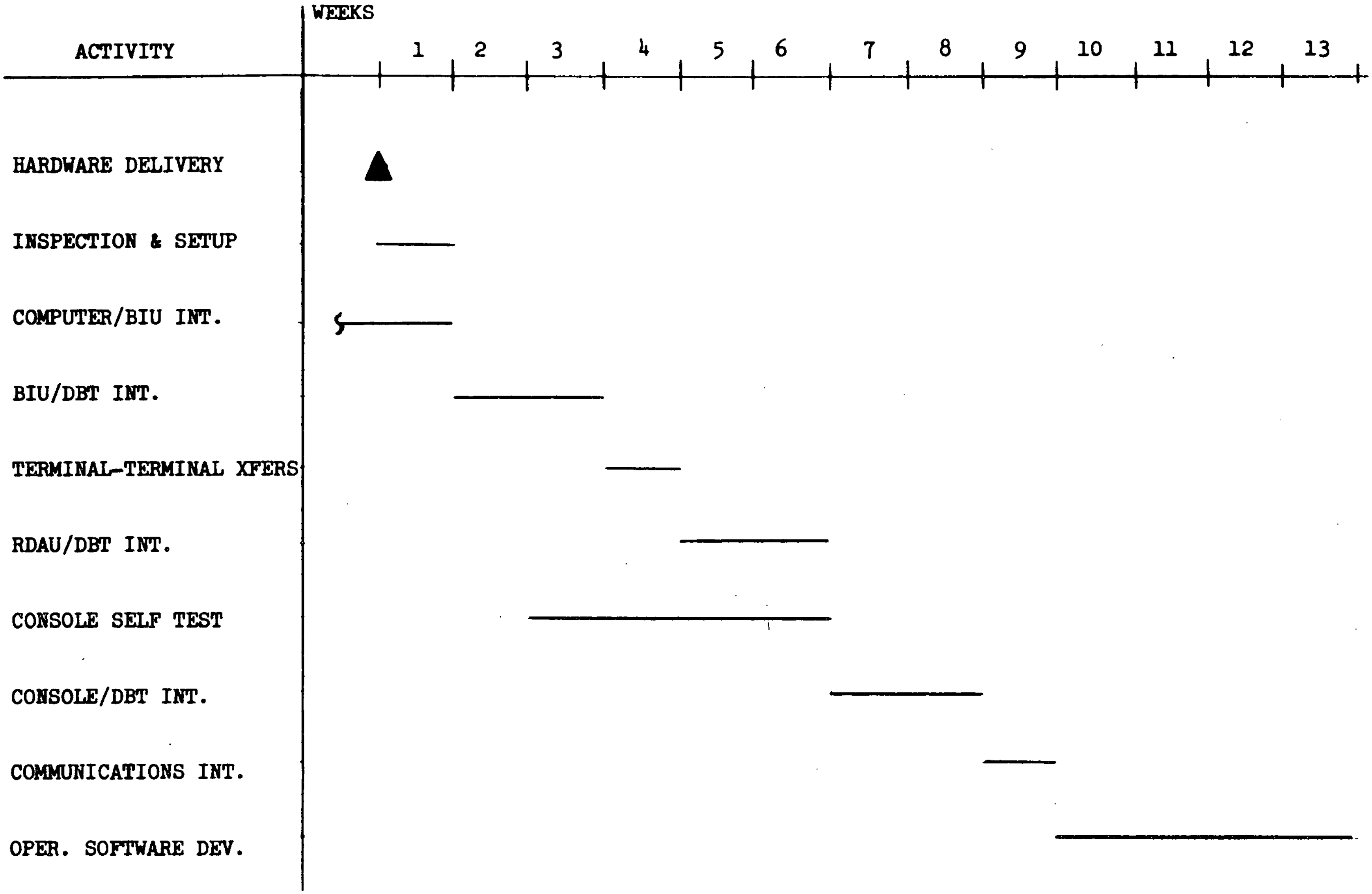


Figure 8-2. Breadboard Development Schedule

The SUMC development and test program will include evaluation of fundamental system organization, including the instruction set and I/O design, in the Space Station environment.

Utilization of the SUMC as the system control computer will require that a new I/O interface unit be designed to provide the SUMC/data bus interface. This device, if designed in accordance with the bus control concepts as defined in this document, will afford direct access to the SUMC main memory and will permit operation of the data bus at its full capacity of 10 megabits/second over multiple frequency-multiplexed channels.

8.2.2.2 Software

Utilization of the SUMC will, of course, require the development of a compatible system of software. The key element in this system will be the executive scheduler module which will provide the overall system control, timing, resource allocation, and similar functions. Two versions of the executive will be required, one for the simplex SUMC configuration and one for the multiprocessor system. A high degree of commonality will exist, however, and in actual fact the simplex system may be considered as a subset of the multiprocessor version. The bulk of the remaining required system software may be utilized by both the simplex and multiprocessor systems. This includes the data bus I/O control, general I/O control, display processing, statement execution, and error detection routines. The utility software, including assembler, compiler, and loader, will also be common to the two configurations, as will the application programs.

8.2.2.3 Data Bus

The initial breadboard digital data bus system may be extended simply by adding additional terminals and interface devices, such as RDAU's, as required. Interfacing the bus with user equipment (such as experiments) will of course require that a compatible interface be provided, either as part of the user equipment or in the form of an adapter.

As mentioned previously, it will be necessary to provide a new data bus/computer I/O interface upon integration of the SUMC as the primary system control computer.

Accommodation of voice, experiment video, and other analog data by the IMS breadboard will require the addition of an analog data bus to the system. The analog bus will interface with the digital bus (via the RDAU) for control, but it will transmit data over a separate cable.

8.2.2.4 Control and Display

The control and display console included in the initial IMS breadboard provides a wide variety of capabilities. Included are alphanumeric, graphic, video, audio, discrete, and analog displays, and input capability consisting of alphanumerical and programmable-function keyboards, light pen, and discrete switches. These should be adequate to accommodate the majority of anticipated breadboard functions. It may be desirable, however, to add to or modify the console as new technology becomes available, or as new requirements develop. It may be desired, for example, to evaluate large screen plasma or light emitting diode displays, or to incorporate special-purpose devices, such as image-processing displays or a hand controller. Some capability has been included to accommodate such growth in the initial console. The bus interface logic contains a spare interface port, and provisions have been made to add a color display capability and a light pen to the video/alphanumeric display. Further growth may be accommodated by additional bus interface equipment.

8.2.2.5 Communications

A logical extension of the K_u-band communications capability will be to add a companion receiver. This would permit a variety of development tests to be conducted. Actual or simulated experiment digital, video, or wide-band analog data could be transmitted to the IMS breadboard from a remote site, for example, and placed on the data bus for transfer to recorders, video monitors, image processing equipment, or other devices, as required. A second alternative would be to simulate video transmission through the use of commercial television signals extracted from a TV receiver IF, remodulated onto the K_u-band carrier, and transmitted to the receiver for demodulation and display.

Other communications capabilities, such as S-band and VHF links, may also be incorporated into the breadboard as requirements arise. All that is required, other than the transmitters and receivers, is that the appropriate data bus interfaces be provided in the form of modems or Data Bus Terminals.

8.2.2.6 User Interfaces

User interfaces may be incorporated into the system as requirements are developed and the necessary hardware becomes available. A wide range of actual or simulated subsystem and experiment interfaces may be accommodated, provided of course that the data bus interface requirements are satisfied and the necessary computer software is developed. Interfaces may be made via the RDAU's, which provide standardized discrete and analog measurement and bilevel command capability; through the digital Data Bus Terminal, which provides a standard 1-megabit serial digital interface; or directly by way of a modem for wide-band analog or high-rate digital data. Detailed specifications for these interfaces may be found in Appendix C of this document.

Section 9
RELIABILITY/MAINTAINABILITY ANALYSIS OF THE IMS DESIGN

Reliability can basically be defined as the probability that the system will operate when and as long as required. It also has the connotation that a device will not only operate but operate properly; i. e., in the case of the data bus, a signal should not only be received, but it should be received without error. Because of the multifaceted nature of the IMS task, a reliability analysis must begin with the basic system concepts, and progress to the reliability embodied in the hardware designs (in this case the breadboard designs). The latter would then include both logic design and physical construction of these units.

Maintainability of the equipment, meaning the design ease of repair and replacement or minimization of repair time, also involves the system concepts as well as the breadboard equipment itself. Therefore, design features of the various IMS contract items will be covered, first at the system level and then at the breadboard hardware level.

9.1 SYSTEM RELIABILITY

The space station system design requires redundant display and control consoles, which are driven by multiprocessors. Each multiprocessor may include up to four processors, three input/output controllers, and fifteen memory modules. Associated with, and controlled by, the multiprocessor is a data bus system consisting of multiple data bus terminals and remote data acquisition units.

The IMS task developed the hardware and software designs for these assemblies to a lower level than had been achievable within the baseline Space Station study. Reliability requirements and techniques resulting from the IMS task are covered in the following paragraphs.

9.1.1 Multiprocessor

The reliability of the multiprocessor is a function of the reliability of its basic building blocks and of its ability to detect faults in the building blocks and take corrective action. Since the design is based on the space ultra-reliable computer (SUMC) the first of these elements is defined. Fault detection relative to the SUMC was found to be beyond the scope of the IMS study, but it was felt that the integral self-test software and hardware could be augmented by using redundant elements to aid in the detection and isolation of faults. It was also recognized that the multiprocessor advantage was lost during reconfiguration, which is a simplex operation. Therefore, a reconfiguration state monitor was defined as special test hardware to detect malfunctions during this period. It features include fail-safe provisions by duplication, provision for testing by the CPU's, ability to detect simultaneous reconfiguration attempts by more than one processor and excessive reconfiguration time, and ability to activate alarms and generate an interrupt upon detection of the foregoing conditions. The device is described in depth in Section 2.1.3.3.

Corrective action in the case of the multiprocessor is in the form of reconfiguration, and a special state (state 3) has been defined for this process (reference: State Control in Section 2.1.3.3). Reconfiguration is directly under executive program control. Features of this program are as follows:

A. Direct CPU-to-CPU and CPU-to-input/output-controller communications augment communications through storage to provide communications redundancy and "instantaneous" communication.

B. Any CPU can execute the executive control program.

C. Through state control and the executive control program, any CPU can remove a malfunctioning CPU from the system.

D. All configuration control can be exercised by the executive control program.

9.1.2 Data Bus

The reliability of the data bus system has been enhanced over and above the baseline Space Station redundant bus design by interfacing each terminal with three modems. This design feature was originally incorporated to increase effective bus rate by allowing simultaneous transmission on multiple subcarriers while insuring that all devices could communicate with each other. However, a secondary result is that an input/output controller can make data transfers over any of the three frequencies (since it has triple redundancy in the input/output channel control, bus channel control, and modem elements) if a secondary or manual means of changing terminal subcarrier channel selection is provided. This modification, although not available in present breadboard units, could easily·be incorporated into subsequent designs.

To insure that signals are received reliably, an analysis was made of error-detection techniques (Section 3.2.1.7). Perhaps of even greater importance, a generous signal-to-noise power margin has been selected. A worst-case coupling analysis was also conducted to insure that power margins are adequate (Section 3.2.3.1).

9.2 SYSTEM MAINTAINABILITY

Concern with system maintainability on the Modular Space Station study has resulted in console designs that are hinged at the top and swing out for access to rear-mounted equipment, the standardization of component construction for ease of accessibility or removal from a rack, modular design for ease of replacement, and standardization of components to allow the use of nondedicated spares. As an example, the benefits accruing from the use of nondedicated spares for the computer subsystems are illustrated by Figure 9-1. Curve A represents a duplex standby arrangement, curve B represents the effect of partitioning a system into ten equal failure rate modules with one dedicated spare per module, curve C results from partitioning a system into ten equal failure-rate modules and providing one nondedicated spare, and curve D is established in a fashion similar to curve C but with two nondedicated spares provided. The decrease in system unreliability resulting from the use of nondedicated spares is obvious.

The IMS special emphasis task, in providing a simplex computer design and extending it to a multiprocessor configuration, has insured that a future Space Station can take advantage of the reliability gains afforded by

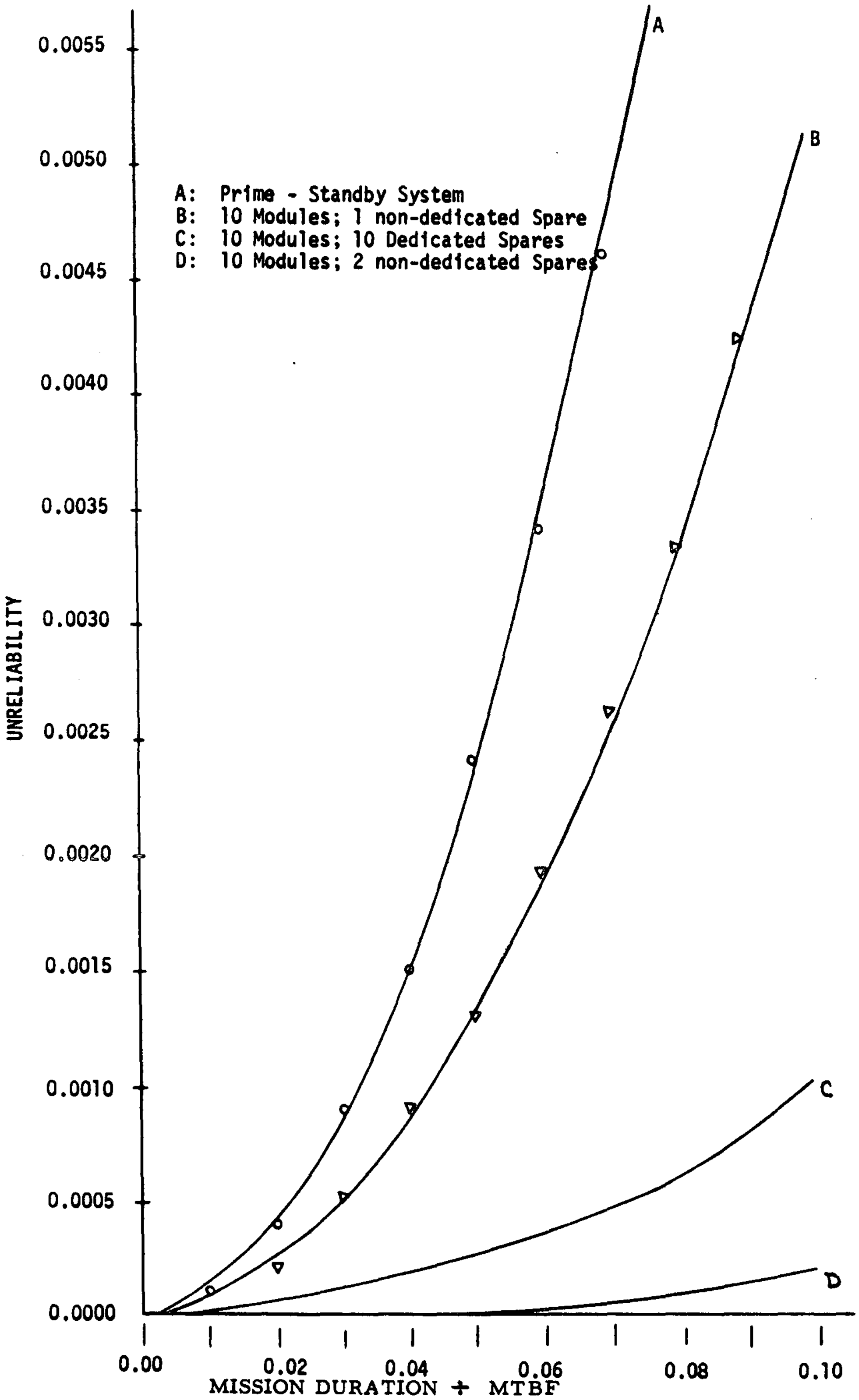


Figure 9-1. Computer System Unreliability

commonality. Maintainability also benefits since fewer spares have to be carried, potentially minimizing repair and replacement time because of the greater familiarity with the single type of equipment that may be achieved.

This same standardization has been carried over into the bus design. All systems interface with the buses through modems. Therefore, the modem has been designed as a separate component with a plug-in capability which allows its integration with the computer I/O, a terminal, or any other system.

The most significant use of commonality in improving maintainability has resulted from the performance of the packaging and maintenance tasks. Guidelines established for the standardization of the packaging design are as follows:

A. The design shall be standardized to the extent that it can accommodate various replaceable assembly groups and the modular support structure can be efficiently used in various vehicle areas.

B. The packaging design of the replaceable assembly shall be capable of accepting multitechnology techniques; i. e., printed-circuit, thick-film, hybrid designs, etc. The equipment shall be designed for maximum flexibility that will insure ability to function with new subsystems and technologies.

C. Commonality is a primary consideration. As a goal, common module structures, a means of standardized mounting, and the like shall be considered.

Having standardized packaging design, maintainability criteria could be imposed with some assurance of its incorporation throughout the system. The criteria itself consists of the provisions enumerated below:

A. Installations shall be designed to permit direct visual and physical access by the crew for replacement or removal of the replaceable electrical assemblies and for disconnecting or connecting related electrical connectors.

B. Access to wire harnesses shall be provided for modifications or repair.

C. Keying on other techniques shall be provided to preclude inadvertent mislocation and improper connector mating of the replaceable assembly.

D. The design shall be compatible with the crew limitations relative to maximum torquing forces, connector extraction forces, adjustment access requirements, assembly form factors, guides, and other constraints.

E. Refurbishable or replaceable captive fasteners shall be used for all equipment that has a planned maintenance capability.

F. The order of precedence for in-orbit maintenance tasks shall be (1) shirtsleeve environment, (2) space suit (IVA), and (3) space suit (EVA). EVA tasks shall be of an emergency nature only.

9.3 BREADBOARD RELIABILITY

The design of the breadboards reflects a concern for their reliability, both at the circuit or logic level and in the physical construction of the hardware.

Steps taken to insure continued operation at the circuit level includes:

A. The use of good commercial parts and practices.

B. A design tailored for the environmental range in which the circuits will be operating.

C. Design based upon worst-case analysis.

D. Use of limited fan-out to reduce device power dissipation.

E. Use of buffers and drivers to preclude overworking of the logic elements.

F. Overvoltage protection (30 volts in the case of RDAU gates).

G. Current limiters where excessive power drains or shorts might occur, such as on command output drivers.

H. Limiting of gate failure effect propogation by "treeing" of the logic.

In the case of power supplies, such special design features as integral cooling fans, overload protection, fuses, and heatsinks have also been incorporated to assure long life and protection from catastrophic failure.

The construction of the special emphasis task equipments provides features that have been selected to enhance both reliability and maintainability. For example, dual in-line packages (DIP's) have been used, which plug into sockets and are wirewrapped instead of soldered. This has minimized the heat applied to the integrated circuits during assembly and improved the ease with which circuits and wiring can be removed for repair or modification.

9.4 MAINTAINABILITY

Other hardware construction features that have improved their maintainability are as follows:

A. Use of standard logic element boards for all plug-in parts to assure ease of accessability.

B. Keying of the logic element boards to prevent inadvertent plug-in error.

C. Use of computer program-controlled wire-wrap interconnections to insure ease of future modification.

D. Placement of components on sliding trays or shelves to provide easy access.

A self-test capability has been provided to assist in maintenance and fault isolation. This includes the following:

A. Display and Control Console

1. Lamp-test switch for simultaneous verification of all lamps and lamp drivers.
2. Self-test switch on the video alphanumeric display to force display of internally stored test patterns.
3. Self-test switch on the voice synthesizer to permit output of selected words or an entire vocabulary.
4. Capability to generate test messages to all display devices, using the programmable data processing capability of the PDS-1 display terminal.
5. Capability to "single-step" the data bus interface logic, using a local test switch.
6. Provision of a status word to allow computer readout of error indications.

B. Data Bus Terminal

1. Capability to single-step the logic, using a local test switch.
2. Provision of a status word to allow computer readout of error indications.

C. Remote Data Acquisition Unit

1. Capability to single-step the logic, using a local test switch.
2. Provision for local display of logic states, using light-emitting diode indicators.
3. Provision of internal self-test voltages for verification of the analog-to-digital converter.
4. Provision of a status word to allow computer readout of error indications.