DTIC FILE COPY

(2)

AFWAL-TR-87-1114

ARCHITECTURE SPECIFICATION FOR PAVE PILLAR AVIONICS

AD-A188 722

ADVANCED INTEGRATION GROUP
SYSTEM AVIONICS DIVISION
AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

DTIC
ELECTE
NOV 19 1987
S D
D

JANUARY 1987

FINAL TECHNICAL REPORT FOR PERIOD
SEPTEMBER 1985 - OCTOBER 1986

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6543

87 11 04 186

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

RONALD SZKODY
ASA Chief Engineer
Advanced Integration Group

JOHN C. OSTGAARD
ASA Program Manager
Advanced Integration Group

FOR THE COMMANDER:

PATRICK L. SISSON, Lt Col, USAF
Deputy Chief
System Avionics Division
Avionics Laboratory

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAS-1, W-P AFB, OH 45433 to help maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

A188722

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-87-1114 SPA90099001A	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Air Force Wright Aeronautical Laboratories 6b. OFFICE SYMBOL (If applicable) AFWAL/AAAS-1	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Wright-Patterson Air Force Base, Ohio 45433-6543	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFWAL 8b. OFFICE SYMBOL (If applicable) AAAS-1	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6543	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 63109F; PROJECT NO. 2734; TASK NO. 01; WORK UNIT ACCESSION NO. 14

11. TITLE (Include Security Classification)
Architecture Specification for PAVE PILLAR Avionics

12. PERSONAL AUTHOR(S)
John Ostgaard, Ron Szkody, Steve Benning, Lt Karl Purdy, Lt Gary Mauersberger, Marvin Soraya

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final Technical Report	FROM Sep1985 TO Oct1986	January 1987	388

16. SUPPLEMENTARY NOTATION
None.

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP	PAVE PILLAR	Common Modules	Integrated Avionics
01	02		System Architecture	Common Hardware	Resource Sharing
01	04		Avionics (Core)	Reconfiguration	Fault Tolerance

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This document establishes the functional requirements for the PAVE PILLAR Systems Architecture. This architecture is specifically targeted for advanced tactical fighters, and in general for all military aircraft applications. The PAVE PILLAR Architecture addresses those functions which could be implemented with common hardware and computer programs to allow adaptation to either air-to-air or air-to-ground missions. This document defines the major elements of the PAVE PILLAR Architecture, the mechanizations for inter-connecting those elements, a set of common modules from which those elements could be constructed, and the operation of the network constructed of these elements.

The PAVE PILLAR core avionics exploits the commonality in air-to-air and air-to-ground missions. The PAVE PILLAR core avionics consists of the following functional areas: (1) Digital Signal Processing, (2) Mission Processing, (3) Vehicle Management Processing, and (4) Avionics Systems Control. The Digital Signal Processing, Mission Processing, and Vehicle Management Processing areas define the enclosing boundaries for resource sharing, sparing, and substitutions. Unique characteristics of each of these areas preclude the utilization of the

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL RON SZKODY	22b. TELEPHONE (Include Area Code) 513-255-4709 22c. OFFICE SYMBOL AFWAL/AAAS-1

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Block 19 continued:

resources across areas for the purpose of functional recovery or reconfiguration.

DTIC
COPY
INSPECTED
6

Accession For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A-1

TABLE OF CONTENTS

Paragraph Number	Title	Page #
1.0	Scope	1
2.0	Applicable Documents	2
2.1	Military Standards	2
2.2	Other Publications	2
3.0	Requirements	3
3.1	Avionics Systems Definition	3
3.1.1	PAVE PILLAR Core Avionics	3
3.1.1.1	Digital Signal Processing Area	3
3.1.1.2	Mission Processing Area	5
3.1.1.3	Vehicle Management Systems Area	5
3.1.1.4	Avionics Systems Control	7
3.1.2	Missions	8
3.1.3	Operational Concept	8
3.1.4	System Diagrams	10
3.1.5	Interface Definition	10
3.1.5.1	Functional Interfaces	10
3.1.5.2	Mission Avionics Multiplex Bus	10
3.1.5.3	Block Transfer Multiplex Bus	12
3.1.5.4	Video Data Distribution System	12
3.1.5.5	Sensor Data Distribution Network	12
3.1.5.6	Sensor Control Network	13
3.1.5.7	Data Exchange Network	13
3.1.5.8	Vehicle Management Systems Multiplex Bus	13
3.1.5.9	Mechanical Interfaces	13

Paragraph Number	Title	Page #
3.1.5.10	Environmental Interfaces	13
3.1.6	Government Furnished Equipment List	14
3.2	Avionics System Characteristics	14
3.2.1	Performance Characteristics	14
3.2.1.1	Fault Tolerance	14
3.2.1.1.1	Hardware Fault Tolerance	15
3.2.1.1.2	Software Fault Tolerance	15
3.2.1.2	Configuration/Reconfiguration Characteristics	15
3.2.1.2.1	Start-Up	15
3.2.1.2.1.1	Vehicle Management System Area Start-Up	17
3.2.1.2.1.2	Digital Signal Processing Area Start-Up	17
3.2.1.2.2	Normal Operations	18
3.2.1.2.3	Mission and Signal Data Processing Failures	19
3.2.1.2.4	Vehicle Management System Data Processor Failure	20
3.2.1.2.5	Sensor Systems Failures	21
3.2.1.2.6	System Mass Memory Failures	22
3.2.2	Physical Characteristics	22
3.2.2.1	Functional Partitioning	23
3.2.2.2	Size	23
3.2.3	Reliability	23
3.2.4	Maintainability	23
3.2.4.1	Mean Time to Repair Critical Functions	23
3.2.4.2	On-Equipment Troubleshooting Time	23
3.2.4.3	Percent Fault Detection	24
3.2.4.4	Percent Fault Isolation	24

Paragraph Number	Title	Page #
3.2.5	Availability	24
3.2.5.1	Sortie Rate	24
3.2.5.2	Abort Rate	25
3.2.5.3	Combat Turnaround Time	25
3.2.5.4	Total Non-Mission Capable for Maintenance of Avionics	25
3.2.5.5	Total Non-Mission Capable for Supply of Avionics	25
3.2.6	Effectiveness Models	25
3.2.7	Environmental Conditions	26
3.2.8	Nuclear Control	26
3.2.9	Transportability	26
3.3	Design and Construction	26
3.3.1	Material, Parts and Processes	26
3.3.2	Electromagnetic Compatibility	26
3.3.3	Nameplate and Product Marking	26
3.3.4	Workmanship	26
3.3.5	Interchangeability	26
3.3.6	Safety	26
3.3.7	Human Engineering	26
3.3.8	Computer Program Design	27
3.3.8.1	Higher Order Language/Run Time Support System	27
3.3.8.2	Software Development System/Design Aids	27
3.3.8.3	Standard Interfaces Requirement	27
3.4	Documentation	31
3.5	Logistics	31

Paragraph Number	Title	Page #
3.5.1	Maintenance Environment	31
3.5.1.1	Maintenance Concept	31
3.5.1.1.1	Organizational Level Maintenance Requirements	31
3.5.1.1.2	Depot Level Maintenance Requirements	32
3.5.2	Supply Support	33
3.5.3	Facilities	33
3.5.4	Cost	33
3.6	Personnel and Training	34
3.7	Functional Area Characteristics	34
3.7.1	Data Processors	35
3.7.1.1	Data Processor Diagrams	35
3.7.1.2	Operating System	35
3.7.1.3	Inter-Module Communications	35
3.7.1.3.1	Inter-Module Communications Bus (PI-Bus)	35
3.7.1.3.2	Test and Maintenance Bus (TM-Bus)	37
3.7.2	System Mass Memory	37
3.7.2.1	System Mass Memory Diagram	38
3.7.2.2	Inter-Module Communications	38
3.7.2.2.1	Inter-Module Communications Bus (PI-Bus)	38
3.7.2.2.2	Test and Maintenance Bus (TM-Bus)	38
3.7.3	Universal Remote Terminals	40
3.7.3.1	System Diagram	40
3.7.3.2	Inter-Module Communications	40
3.7.3.2.1	Inter-Module Communications Bus (PI-Bus)	40
3.7.3.2.2	Test and Maintenance Bus (TM-Bus)	42

Paragraph Number	Title	Page #
3.7.4	Mission Avionics Multiplex Bus	42
3.7.4.1	Topology	42
3.7.4.2	Protocol	43
3.7.4.3	Control	43
3.7.4.4	Interface Characteristics	43
3.7.4.5	Physical Characteristics	43
3.7.4.6	Fault Recovery	43
3.7.4.7	Nuclear Hardness	44
3.7.5	Block Transfer Bus	44
3.7.5.1	Topology	44
3.7.5.2	Protocol	44
3.7.5.3	Control	45
3.7.5.4	Interface Characteristics	45
3.7.5.5	Physical Characteristics	45
3.7.5.6	Fault Recovery	45
3.7.5.7	Nuclear Hardness	45
3.7.6	Vehicle Management System Multiplex Bus	45
3.7.6.1	Topology	46
3.7.6.2	Protocol	46
3.7.6.3	Control	46
3.7.6.4	Interface Characteristics	46
3.7.6.5	Physical Characteristics	47
3.7.6.6	Fault Recovery	47
3.7.6.7	Nuclear Hardness	47
3.7.7	Common Signal Processor (CSP)	47

Paragraph Number	Title	Page #
3.7.7.1	Common Signal Processor Diagram	47
3.7.7.2	Inter-Module Communications	47
3.7.7.2.1	Inter-Module Communications Bus (PI-Bus)	50
3.7.7.2.2	Data Flow Network	50
3.7.7.2.3	Test and Maintenance Bus (TM-Bus)	50
3.7.8	Sensor Data Distribution Network	51
3.7.8.1	System Diagram	52
3.7.9	Data Exchange Network	52
3.7.9.1	System Diagram	52
3.7.10	Video Data Distribution Network	52
3.7.10.1	System Diagram	55
3.7.11	System Executive	55
3.7.12	Kernel Executive	57
3.7.13	Distributed Executive	60
4.0	Quality Assurance Provisions	60
5.0	Preparation for Delivery	60
6.0	Notes	62
6.1	Common Module Set	62
6.1.1	VHSIC 1750A Processor Module (V1750A)	63
6.1.2	Avionics Bus Interface (ABI) Module	65
6.1.3	Bulk Memory Module (BMM)/Non-Volatile Bulk Memory Module (NVBMM)	65
6.1.4	MIL-STD-1553B I/O Module	65
6.1.5	Dedicated Interface Modules	68
6.1.6	Floating Point Processing Element (FPPE) Module	69
6.1.7	Global Memory (GM) Module	71

Paragraph Number	Title	Page #
6.1.8	Data Network Element (DNE) Module	73
6.1.9	Sensor Interface (SI) Module	74
6.1.10	Element Supervisor Unit (ESU) Module	77
6.1.11	Timing and Control Generator (TCG) Module	79
6.1.12	I/O Terminator Module	80
6.1.13	DC - DC 5 Volt Power Converter Module	81
6.1.14	DC - DC Multi-Voltage Power Converter Module	82
6.1.15	Power Conditioner Module	83
6.1.16	Fixed Point Vector Processing Element (VPE) Module	84
6.1.17	Electronic Warfare-Oriented Processing Element Modules	85
6.1.18	Biphase Correlator Processing Element (BCPE) Module	86
6.1.19	MIL-STD-1760 Stores I/O Module	86
6.1.20	Video Module	88
6.1.21	Display Module	88
6.1.22	Network Switch Module (NSM)	90
6.1.23	Key Generator Module (KGM)	92
6.2	Packaging	93
6.2.1	Equipment Enclosure	93
6.2.2	Line Replaceable Module	95
6.2.3	Damage Tolerance	95
6.3	Executive to Applications Interface	96
6.3.1	Applications Interface with the Executive	96

Paragraph Number	Title	Page #
6.3.2	Executive Interfaces with Applications Tasks	96
6.3.3	Usage of Executive/Applications Interfaces	97
6.4	Non-Homogeneous Processing Resources	98
6.4.1	Standalone Subsystem	98
6.4.2	Shared Data Processing Subsystem	99
6.4.3	Shared Data and Signal Processing Subsystem	101
6.4.4	Incomplete Sensor Interconnectivity	103
6.4.5	Non-Homogeneous Data Processors	103
6.4.6	Non-Homogeneous Signal Processors	103
6.5	Acronyms	105

List of Tables

Number	Title	Page #
3.1.1.3-1	Vehicle Management System Functions	6
3.7.7-1	Typical Signal Processing Functions	48
6.1-1	Baseline Common Module Set	64

List of Figures

Number	Title	Page #
3.1-1	Avionics System Functional Organization	4
3.1.1.4-1	Functional Partitioning of System Software	9
3.1.4-1	Common Architecture	11
3.3.8.3-1	Classes of Interfaces	29
3.7.1-1	Data Processor Modular Structure	36
3.7.1-2	Vehicle Management System Data Processor Modular Structure	36
3.7.2.1-1	System Mass Memory Modular Structure	39
3.7.3-1	Universal Remote Terminal Module Structure	41
3.7.7.1-1	Common Signal Processor Hardware Architecture	49
3.7.8-1	Sensor Data Distribution Network	53
3.7.9-1	Data Exchange Network	54
3.7.10-1	Video Data Distribution Network	56
3.7.11-1	System Executive	58
3.7.12-1	Kernel Executive	59
3.7.13-1	Distributed Executive	61
6.1.1-1	1750A Processor Module	66
6.1.2-1	Avionics Bus Interface Module	66
6.1.3-1	Bulk Memory Module	67
6.1.4-1	MIL-STD-1553B I/O Module	67
6.1.5-1	Dedicated Interface Module	68
6.1.6-1	Floating Point Processing Element Module	70
6.1.7-1	Global Memory Module	72
6.1.8-1	Data Network Element Module	74

Number	Title	Page #
6.1.9-1	Sensor Interface Module	76
6.1.10-1	Element Supervisor Unit Module	78
6.1.11-1	Timing and Control Generator Module	79
6.1.12-1	I/O Terminator Module	80
6.1.13-1	DC - DC 5 Volt Power Converter Module	81
6.1.14-1	DC - DC Multi-Voltage Power Converter Module	82
6.1.15-1	Power Conditioner Module	83
6.1.16-1	Fixed Point Vector Processing Element Module	84
6.1.17-1	EW Processing Element Modules	85
6.1.18-1	Biphase Correlator Processing Element Module	87
6.1.19-1	MIL-STD-1760 Stores I/O Module	87
6.1.20-1	Video Module	89
6.1.21-1	Display Module	89
6.1.22-1	Network Switch Module	90
6.1.22-2	Network Switch Utilization	91
6.1.23-1	Key Generator Module	92
6.4.1-1	Standalone Subsystem Interfaces	100
6.4.2-1	Shared Data Processor Subsystem Interfaces	100
6.4.3-1	Shared Data and Signal Processing Subsystem Interfaces	102
6.4.4-1	Incomplete Sensor Interconnectivity	104

Appendices

Number	Title	Page #
Appendix A	General Specification for 3/4 ATR Module	A
Appendix B	VHSIC Phase 2 Interoperability Standards-PI Bus	B-1
Appendix C	VHSIC Phase 2 Interoperability Standards-TM Bus	C-1
Appendix D	High Speed Data Bus Protocol Specification	D

1.0 SCOPE

This document establishes the functional requirements for the PAVE PILLAR Systems Architecture. This architecture is specifically targeted for advanced tactical fighters, and in general for all military aircraft applications. The PAVE PILLAR Architecture addresses those functions which could be implemented with common hardware and computer programs to allow adaptation to either air-to-air or air-to-ground missions. This document defines the major elements of the PAVE PILLAR Architecture, the mechanizations for interconnecting those elements, a set of common modules from which those elements could be constructed, and the operation of the network constructed of these elements.

This document describes a fully integrated architecture concept which includes common data and signal processors. Section 6.4 addresses the issue of non-homogeneous processing resources.

Appended to this document is a set of specifications which represent current physical and functional interface specifications for the PAVE PILLAR architecture as implemented by Advanced Development Model (ADM) technology programs. As such, these specifications may by revised and expanded where necessary when applied to Full Scale Engineering Development (FSED) programs.

2.0 APPLICABLE DOCUMENTS

2.1 Military Standards

MIL-STD-1553B, Aircraft Internal Time Division Command/Response Multiplex Data Bus

MIL-STD-1750A 16 Bit Computer Instruction Set Architecture

MIL-STD-1760 Aircraft Store Electrical Interconnections System

MIL-STD-1815A Ada Programming Language

MIL-STD-483A Configuration Management Practices for Systems, Equipment, Munitions, and Computer Programs

MIL-STD-490A Specification Practices

DOD-STD-2167 Defense System Software Development

2.2 Other Publications

NASCIM 5100A (C) Comprising Emanations, Laboratory Test Requirements, Electromagnetic

NACSEM 5112 (S-NF) Non-Stop Evaluation Techniques

NACSEM 5201 (C) Tempest Guideline for Equipment/System Design

3.0 REQUIREMENTS

3.1 Avionics Systems Definition

An avionics system is a suite of equipment integrated into the aircraft to enable the pilot to accomplish the assigned missions of a tactical fighter aircraft weapon system. The avionics system is divided into four functional areas: 1) Sensor/Subsystems, 2) Digital Signal Processing, 3) Mission Processing, and 4) Vehicle Management Processing.

Figure 3.1-1 shows the components and the communication network for the Vehicle Management Processing Area, Digital Signal Processing Area, and the Mission Processing Area.

3.1.1 PAVE PILLAR Core Avionics

The PAVE PILLAR core avionics exploits the commonality in air-to-air and air-to-ground missions. The PAVE PILLAR core avionics consists of the following functional areas: 1) Digital Signal Processing, 2) Mission Processing, 3) Vehicle Management Processing, and 4) Avionics Systems Control. The Digital Signal Processing, Mission Processing, and Vehicle Management Processing areas define the enclosing boundaries for resource sharing, sparing, and substitutions. Unique characteristics of each of these areas preclude the utilization of the resources across areas for the purpose of functional recovery or reconfiguration.

3.1.1.1 Digital Signal Processing Area

The Digital Signal Processing Area provides the resources to perform the digital signal processing for radar, electronic warfare, image and CNI processing. Resources will be provided to route sensor data from the sensor/subsystem to the appropriate signal processing element. Resources will be

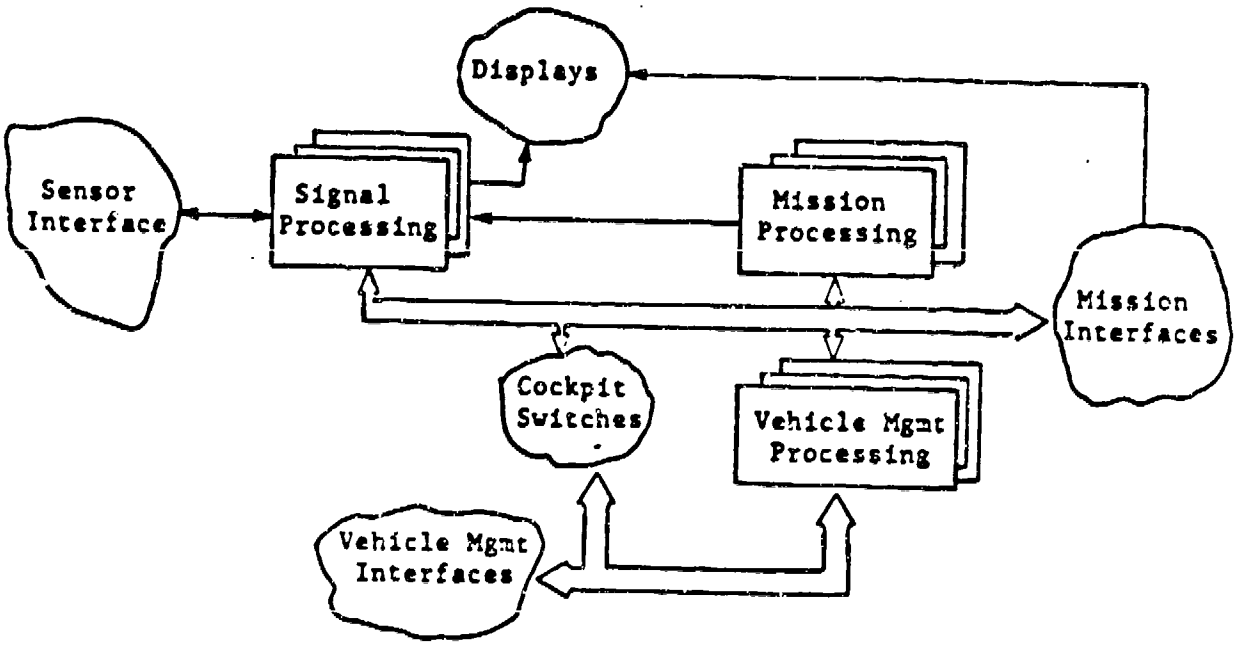


Figure 3.1-1 AVIONICS SYSTEM FUNCTIONAL ORGANIZATION

provided to exchange data between signal processing elements. The major components of the Signal Processing Area are a set of common signal processors, a sensor data distribution network, a sensor control network, a data exchange network, and a video data distribution system.

3.1.1.2 Mission Processing Area

The Mission Processing Area provides the resources to perform mission and system management such as fire control, target acquisition, navigation management, defense management, stores management, TF/TA/OA functions, and crew station management. The Mission Processing Area controls the reconfiguration of the Digital Signal Processing Area and determines the routing of the data from the sensor/subsystem to the signal processing elements. The Mission Processing Area collects the health and status of all avionics components for maintenance history and maintains a record of mission functional capability. The components of the Mission Processing Area are mission data processors, mission avionics multiplex bus, block transfer multiplex bus, system mass memory, stores management system, and a collection of interfaces to the mission avionics multiplex bus.

3.1.1.3 Vehicle Management Systems Area

The Vehicle Management Systems Area provides the resources to support the fundamental flight and airframe related control functions listed in Table 3.1.1.3-1. The components of the Vehicle Management System Area are vehicle management system data processors, control/display interfaces, flight sensor/actuation interfaces, electrical power control interfaces, engine control interfaces, and utility systems interfaces.

TABLE 3.1.1.3-1
VEHICLE MANAGEMENT SYSTEM FUNCTIONS

Flight Control
Inlet Control
Propulsion Control
Vector Thrust Control
Air Data Measurement
Aircraft Inertial Measurement
Electrical Power Control
Utility Systems

- Fuel Measurement and Transfer
- Fuel Inerting System
- Environmental Measurement and Control
- Life Support Control
- Crew Escape
- Hydraulic System
- Landing Gear
- Auxiliary Functions (Refueling, Landing Lights, etc.)

3.1.1.4 Avionics Systems Control

Control of the PAVE PILLAR avionics is vested in a distributed control architecture providing for a maximum efficiency in resource utilization, mission effectiveness, and commonality of control software across major processing units. Major control functions must include:

1) Initialization and system start-up/restart

2) Assignment of application software task to processing resources (software configuration and reconfiguration computing resources management)

3) Sequencing and synchronization of related software tasks

4) Management of sensor and other device resources with respect to mission objectives, mode/task management and software parameters

5) Interpretation of response to, and integration of, human control into the system functionality

6) Collection, maintenance, and reporting of system hardware and software status, and operational functionality

7) Response to hardware and software failure detection to preserve mission effectiveness

8) Flight control change management and response

9) Reintegration of recovered hardware and software functions

10) Assurance of a distributed data base consistency and integrity, and management of access to that shared data

11) Preservation/collection of data required for continuity of system functionality across failure recovery points

12) Management of communications access to assure optimal use of communication resources and correct addressing of data messages

13) Assurance of the security of classified data.

The avionics systems control functions of the operating system shall be partitioned into three elements: 1) The system executive which will provide the monitoring of system state and the reconfiguration based upon mission requirements and detected system failures; 2) the distributed executive which will provide for decentralized system control in each processor; 3) the kernel executive which will provide those operating system functions which are common to all processors. Figure 3.1.1.4-1 depicts the interrelationship of the three elements.

3.1.2 Missions The PAVE PILLAR core avionics will support both air-to-air and air-to-ground tactical missions.

3.1.3 Operational Concept

The PAVE PILLAR architectural concept was developed to support aircraft operations from deployed locations with a minimum of support. This architecture supports the resource sharing of core data and signal processing resources and is

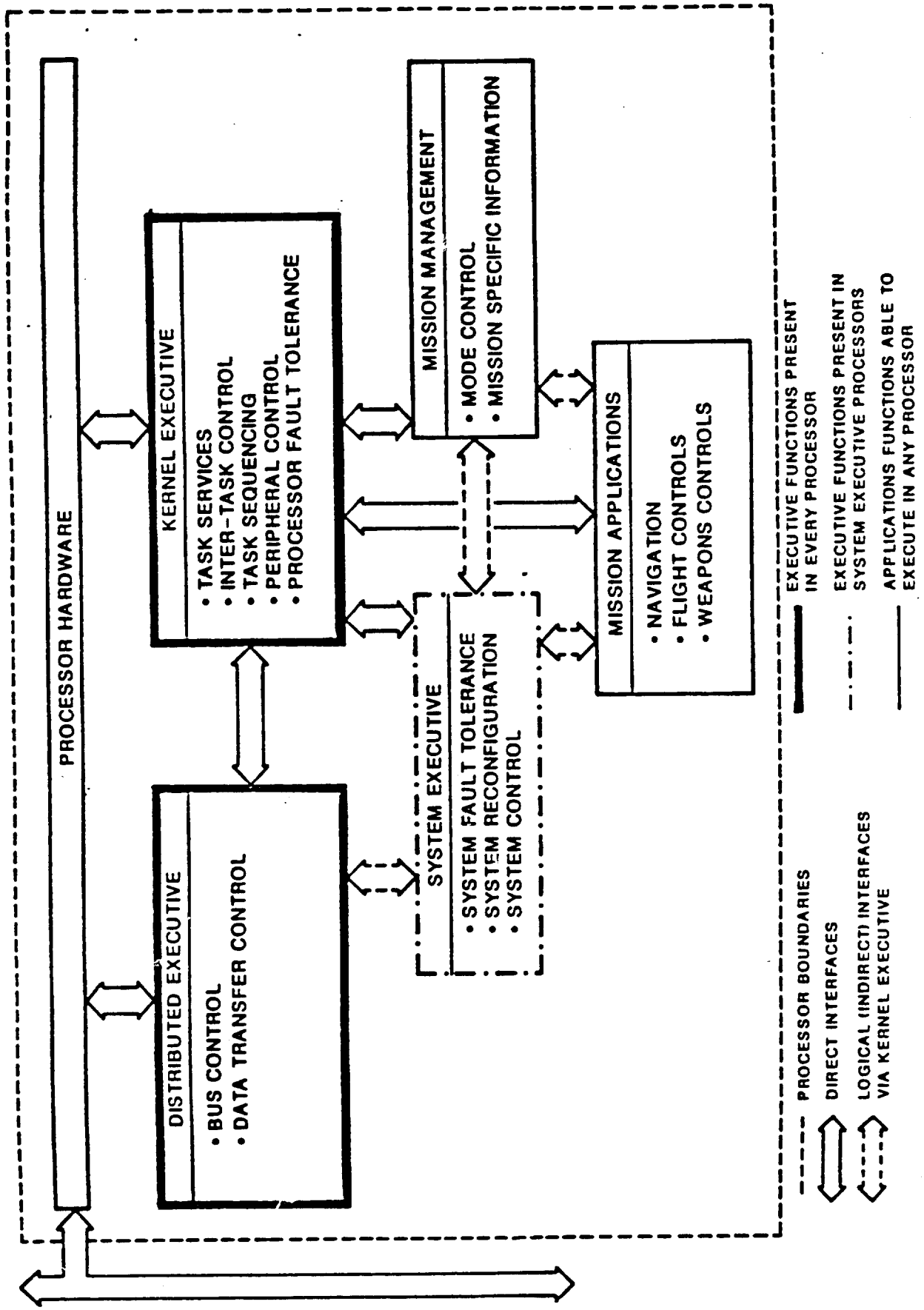


Figure 3.1.1.4-1 FUNCTIONAL PARTITIONING OF SYSTEM SOFTWARE

constructed of a set of common modules that specifically support a two-level maintenance concept. This architecture supports high degrees of systems availability and reliability. This is accomplished through the application of spare signal and data processing resources at the system level so that backup services are provided when the primary sources fail. In addition, the architecture supports graceful degradation in that when spare resources are exhausted remaining resources can be assigned to the highest priority functions on a mission basis.

3.1.4 System Diagrams

The PAVE PILLAR core avionics systems and their common components are shown in Figure 3.1.4-1. A detailed description of each of the common components is presented in Paragraph 3.7.

3.1.5 Interface Definition

3.1.5.1 Functional Interfaces

The PAVE PILLAR core avionics interfaces with the sensor/subsystems, the crew station avionics, and the weapons. Interface control documents will be developed for each sensor/subsystem that interfaces with the PAVE PILLAR core avionics. The preferred concept is that all System Element Level Interfaces be implemented using Fiber Optic Technology and make maximum use of common interface definitions.

3.1.5.2 Mission Avionics Multiplex Bus

The Mission Avionics Multiplex Bus connects all the mission data processors, vehicle management system data processors, signal processors, sensor data distribution network,

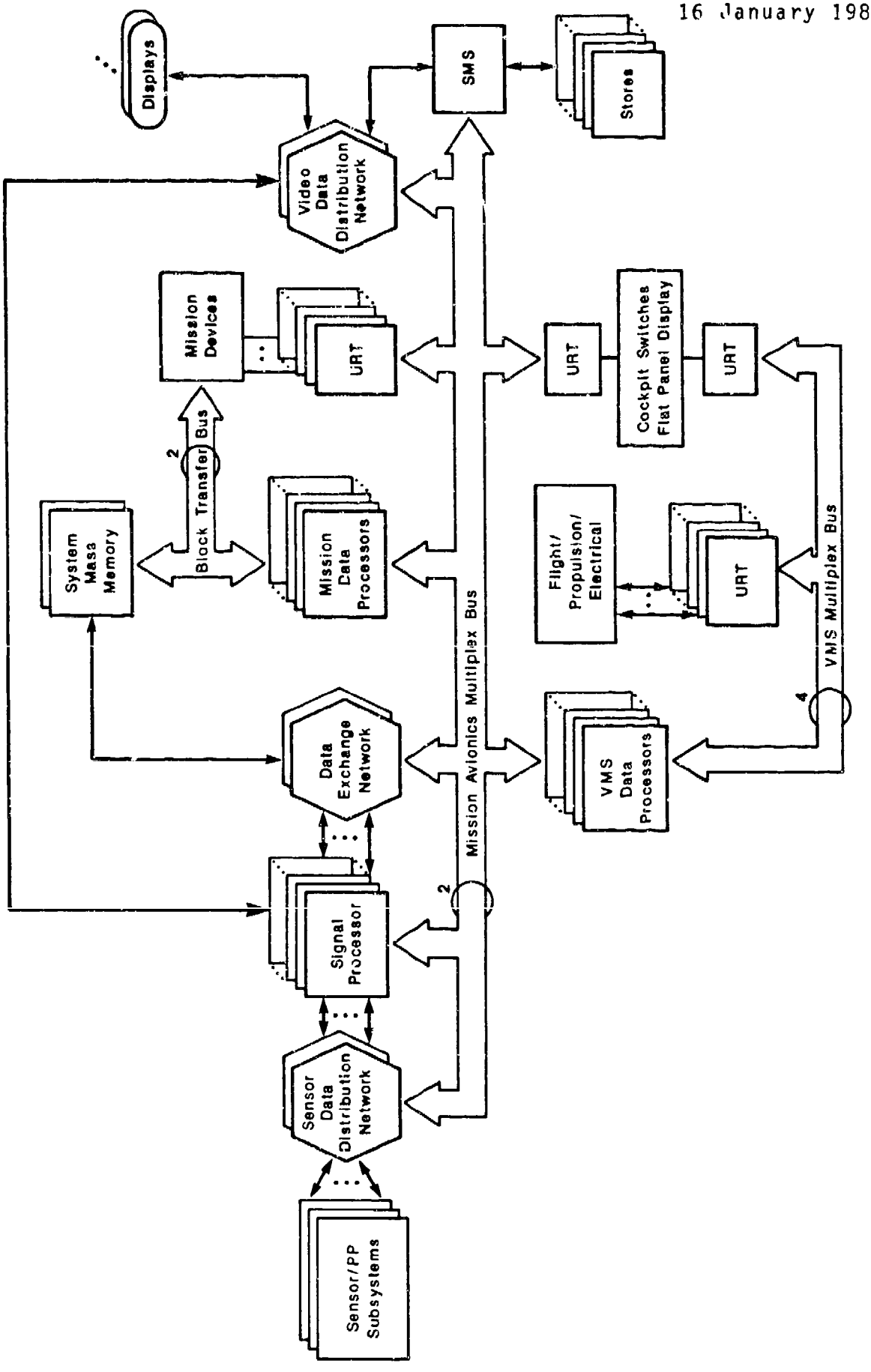


Figure 3.1.4-1 COMMON ARCHITECTURE

data exchange network, video data distribution network and mission interface terminals.

3.1.5.3 Block Transfer Multiplex Bus

The Block Transfer Multiplex Bus is connected to all the mission data processors, the system mass memory, and data transport units. The Block Transfer Multiplex Bus shall be used to load volatile processor memories and transfer large blocks of data.

3.1.5.4 Video Data Distribution System

The Video Data Distribution System integrates the video distribution system on two levels. The two levels are the internal Video Data Distribution Network (VDDN) and the external (or stores) video distribution. The VDDN shall provide the interface between the cockpit displays, the stores management, and the signal processors. The VDDN is controlled through the Mission Avionics Multiplex Bus. The external video distribution shall route the video lines, defined in the MIL-STD-1760 Stores Interface Requirements, to the stores management system where any of the video signals may be selected by the stores interface to be connect d to the internal video distribution system.

3.1.5.5 Sensor Data Distribution Network

The Sensor Data Distribution Network (SDDN) shall allow any one sensor's data to be distributed to any one signal processor. The SDDN is controlled through the Mission Avionics Multiplex Bus.

3.1.5.6 Sensor Control Network

The Sensor Control Network shall provide the communication paths from any signal processor back to the sensor subsystem it is controlling and receiving data from. The Sensor Control Network is part of the SDDN routing network and is also controlled through the Mission Avionics Multiplex Bus.

3.1.5.7 Data Exchange Network

The Data Exchange Network shall provide the communication paths between the system mass memory, the signal processing areas, and COMSEC/TRANSEC controllers. The Data Exchange Network is controlled through the Mission Avionics Multiplex Bus.

3.1.5.8 Vehicle Management Systems Multiplex Bus

The Vehicle Management Systems Multiplex Bus shall interface the Vehicle Management Systems data processors with flat panel display/cockpit switches and integrates the flight propulsion and electrical subsystems.

3.1.5.9 Mechanical Interfaces

The mechanical interfaces with PAVE PILLAR core avionics will be determined by the contractor and documented in the mechanical interface document.

3.1.5.10 Environmental Interfaces

The PAVE PILLAR core avionics will provide the specified performance when interfaced with an aircraft environmental control system. The environmental control for the avionics will be determined by the contractor and documented in the interface control documents.

3.1.6 Government Furnished Equipm nt List (None)

3.2 Avionics System Characteristics

3.2.1 Performance Characteristics

3.2.1.1 Fault Tolerance

State-of-the-art techniques in both hardware and software designs shall be used to establish a high level of system tolerance to both hardware and software failures. The essential capabilities to be provided include the:

1) Ability to detect, correct, or compensate for soft errors induced by transient hardware or environmental anomalies. For example, memory bit errors, data transmission errors, etc.

2) Ability to detect, locate and isolate non-automatically correctable hardware and software failures so that graceful recovery reconfiguration may take place without loss of data or function. Where the nature of the failure necessitates recovery to a degraded mode, any data loss or functional loss must be identifiable and under the control of the system. In the maintenance mode, detection and localization of faulty hardware to a line replaceable module must be provided.

3) Ability to recover from soft and hard failures with minimum disruption to, or intervention of, the user and maximum preservation of system function optimized toward the objectives of the current mission.

4) Ability to contain faults to prevent the spread of system or data contamination.

3.2.1.1.1 Hardware Fault Tolerance

The hardware techniques that shall be applied to providing the required detection capability and hardware remedies shall include the use of redundant busing, spare and redundant common modules, hot standby spare and dual redundant processors, self-checking hardware circuitry and built-in test.

3.2.1.1.2 Software Fault Tolerance

Software fault tolerance shall be provided for each processor. Software fault tolerance shall be provided through the use of dynamic error handling techniques. These techniques involve the use of operational software code to detect, confine, and correct software data and timing errors during systems operation, in order to limit the duration of a software failure or to prevent a software error from causing a software failure, if possible. A software failure is an event caused by software error which results in service interruption. A software data error is an item of information generated by software fault which leads to a service interruption if left uncorrected. A software timing error is a situation caused by a software fault which prevents an item of information from being provided within a specified time or sequence. Dynamic error handling techniques shall be specified as appropriate for critical or complex logic, critically timed services, all interfaces including software/hardware and hardware/hardware input data, and critical data values.

3.2.1.2 Configuration/Reconfiguration Characteristics

3.2.1.2.1 Start-up

In high level terms, the bulk of the system level control will be centered in the mission processors designated as the system executive and its redundant backup. Since these

processors, as hardware, shall be identical to all other processors, the determination of the specific processors to act as system supervisors may be flexibly accomplished. Executive software shall detect the absence of supervisors in the system upon start-up and respond by adopting the supervisor's role. The processors to adopt this role will be the ones that first acquire access to the Block Transfer Multiplex Bus and, thereby, to the System Mass Memory in the absence of supervisors. These processors shall proceed with self tests and load the mission management and system level executive software. This software shall be unique to the processors that fulfill the supervisor's role. Subsequently, as they gain access to System Mass Memory, all other Mission Data Processors shall execute self tests and shall be loaded to establish an initial mission software configuration in accordance with the functionality dictated by the mission plan and data base resident in System Mass Memory. The system supervisor shall verify that the required status is reached by the full complement of mission processors and report positively to the crew upon successful completion of this stage. In the event that the required status is not reached within a reasonably defined time period, a fault condition shall be reported. If automatic recovery from the deficient system condition is appropriate, the supervisor shall initiate recovery procedures. Once the Mission Data Processing configuration is established, the supervisor shall begin gathering status data on all system elements accessible on the Mission Avionics Multiplex Bus. As establishing communications reveals that these elements have become operational (as a consequence of their start-up procedure), updated status displays will be generated and transmitted to the cockpit. The operational availability and status of each PAVE PILLAR element or interface subsystem shall be compared against the mission resource requirement. If full mission capability is reached, this fact shall be reported. Until that occurs, a report that mission resources are lacking will be made available to the crew.

3.2.1.2.1.1 Vehicle Management System Area Start-up

In a completely autonomous cycle, the Vehicle Management Systems Area, under the control of the mission independent Read Only Memory (ROM) software, shall establish itself carrying out self test upon start-up and verifying all VMS bus devices and interfaces. Status reporting by the VMS processor shall be made independently to the cockpit and to the Mission Processing Area. Upon VMS processor request, any system data required for integrated systems operation and communication shall be provided to the VMS processor by the responsible Mission Data Processor(s). The VMS shall be capable, however, of entirely autonomous operation and shall exercise complete control over VMS to Mission Processing Area communications as well as over its own interfaces to the cockpit subsystem. No software load shall be required since the necessary software exists in Read Only Memory.

3.2.1.2.1.2 Digital Signal Processor Area Start-up

The control processors within the signal processors shall be responsible for control of signal processor testing and internal module operational status gathering which it shall report to the system supervisor. When the supervisor has determined an operational status for each signal processor, it shall make, based upon mission requirements, an assignment of signal processor to sensor/subsystem signal sources. The supervisor shall send appropriate control data to the sensor signal switching network controller to set up the determined configuration and the signal processor shall be instructed as to their required software loads which they shall proceed to acquire from System Mass Memory. Once loaded, each signal processor shall establish and test communications with its assigned sensor subsystem on both control and data communications networks. The status of these operations shall be reported to the system supervisor by each signal processor as data for the

supervisor's evaluation of mission readiness and/or recovery initiation. Appropriate protocol and time-out thresholds to allow detection by the supervisor of failures in the signal processing area during these activities shall be established.

3.2.1.2.2 Normal Operations

Any operational state that is stable (that is, fault processing, recovery processing, or mode change-over processing are not active) shall be termed "normal operations." Thus, normal operations may include degraded but stabilized modes of operation. During normal operations, the role of the system supervisor diminishes to monitoring a software/hardware configuration and status of Mission Data Processor tasks and Mission Avionics Multiplex Bus devices with respect to mission requirements (for example, to detect an event or time requiring flight mode changes) and to performing the processing of crew control and information requests in order to insure their validity and consistency with the current system mission, configuration and status. Mission authorized manual control/ override of automated avionics functions shall be made possible at all times. The responsibility for any configuration or reparameterization required to support manual operation rests with the system supervisor. The sequencing and synchronization of major steps in carrying out a mission, for example mission mode changes, sensor systems initialization, parameterization, etc. shall be initiated and directed by the supervisor. The remaining mission processors, once task definition loads and parameters have been established in response to system supervisor's directives and/or mission data base content, shall carry out the on-going control functions of task sequencing and synchronization through task-to-task communication services and interprocessor communication services.

3.2.1.2.3 Mission and Signal Data Processing Failures

Notification of a processor or interprocessor communications hardware fault will first be made to the executive function in the processor. The complete failure of a processor or device on the Mission Avionics Multiplex Bus shall be detected by at least the system supervisor through the data bus protocol, revealing the lack of anticipated transmission. This detection is initially handled at the executive level in that portion of the executive responsible for communications access.

In the case of a complete processor failure, the system supervisor shall initiate or allow the system to initiate in distributed implementations recovery by that procedure which, either by preset design, or by an algorithmic determination, is optimal for the preservation of mission functions and utilization of resources toward the goal of maintaining the maximum probability of mission success.

The required recovery action shall consist of hardware reconfiguration, software reconfiguration or a combination of the two. Upon loss of a complete processor device or communications bus, the System Executive shall determine where there is available redundant or standby hardware to simply take over the failed element's function. If so, the hardware reconfiguration shall take place at the executive level with the appropriate notifications to the high level functions in all processors. In the event such a simple solution is not possible, the system supervisor determines the required hardware/software reconfiguration needed and initiates communications to accomplish the task. In this way, changes in mode, degradation considerations, crew notification and responses, and a complete knowledge of mission requirements and objectives may be combined to best determine a solution to the problem presented by the failure. To the fullest extent safely possible, the solution to the reconfiguration problem shall be

carried out automatically by the system. Notification of the reduced capability implications on mission objectives and aircraft safety shall be made available to the crew.

When a Signal or Mission Data Processor fault is detected local to the processor which may be fully recovered within the processor, control and responsibility for recovery and reporting to the system supervisor of the required state change shall rest with the Distributed Executive and applications function and, with the exception of notification, shall be transparent to the remainder of the system. A fault that is determined to simply reduce, but not eliminate, the capability of the processor shall be reported to the supervisor where reconfiguration determination shall then reside.

In those cases where reconfiguration of software is undertaken in Mission Data and Signal Processors, those processors acquiring an altered complement of functions shall, upon instruction as to the new requirements, initiate appropriate total and partial software loads from System Mass Memory. The system supervisor must verify the new configuration upon completion of the reconfiguration and synchronization and the re-establishment of interrupted mission and sensor data processing. Wherever possible, functions not involved in the reconfiguration shall continue uninterrupted.

3.2.1.2.4 Vehicle Management System Data Processor Failure

Flight essential reliability requirements for the VMS stem from the critical nature of the majority of functions provided there such as flight controls actuation, propulsion control, electrical distribution control, etc. The response to failure of a VMS data processor must therefore be swift and certain resulting in an absolute minimum, if any, disruption of service. The potentially disastrous results of erroneous control commands also require that integrated hardware and

software measures be taken to insure against software and temporary error conditions.

Detection of either hard or soft failures in a processor shall result in immediate switchover to a pooled spare VMS data processor. The responsibility for this response shall be vested in the executive software. The number of levels of redundancy shall be determined to meet the flight essential reliability requirements. In cases, if any, where changes in function served by VMS data processor are dictated by failure condition, this shall be accomplished by activation of non-volatile software routines, the entire software set being resident in each Non-Volatile 1750A Processor Module. To guard against loss of control in the VMS, the control mechanization shall be highly distributed. If the computing resources remaining after one or more failures are sufficient, if utilized properly, to provide the critical control functions, the VMS shall not fail as a result of the loss of software system control.

Failure of devices, device interfaces or buses shall be detected and initially processed by the executive. Necessary notification to application software for failure conditions shall be initiated as a result of executive processing.

3.2.1.2.5 Sensor Systems Failures

Control of the sensor/subsystems both in terms of communications and in terms of control command generation shall reside in the control processor of the responsible signal processor. All systems originated sensor control command shall be submitted to the appropriate signal processor's control processor for passage to the sensor. Within the signal processor, sequencing of commands for optimization, resolution of conflicting commands, etc. shall be carried out and

appropriate notification sent to the relevant processor. Sensor failures shall be detected by the Signal Processor Executive. In the event failure recovery is not possible, the Signal Processor fault handling software shall notify the system supervisor of the fault details and shall notify all processors expecting data from the signal processor that the sensor is unavailable. Appropriate responses to the sensor loss shall be taken by application software in the notified processors. The determination of system level response to sensor loss in the overall control/direction of the response shall be carried out by the system executive.

If a sensor system has recovery capability (for example, switchover to a backup sensor, that may cause temporary disruption), established protocol within the affected signal processor shall cause the signal processor to become aware of the condition. The signal processor application software, upon being notified, shall carry out any required notification of the dependent processing modules and notify the system executive of the disruption. Otherwise, the response to this level of sensor fault shall reside in the signal processor.

3.2.1.2.6 System Mass Memory Failures

Limited loss of memory capability without loss of reloading capability shall be tolerated by the system but shall be transparent to all users of memory access mechanizations. Upon failure of the active System Mass Memory, the system shall automatically, with transparency to the remainder of system, redirect all memory accesses to the duplicate System Mass Memory. Status indication shall inform the system supervisor of the change of state.

3.2.2 Physical Characteristics

3.2.2.1 Functional Partitioning

The PAVE PILLAR core avionics shall be functionally partitioned into plug-in line replaceable modules. The line replaceable modules may be installed in line replaceable assemblies and/or integrated racks.

3.2.2.2 Size

The PAVE PILLAR common line replaceable module shall be mechanically compatible with three-quarter ATR size. (See Appendix A - General Specification for 3/4 ATR Module.)

3.2.3 Reliability

The PAVE PILLAR avionics system shall be designed to support a mean time between critical failure of 70 hours. Mean time between critical failure is the average operating time between critical failure. A critical failure is a failure of an essential subsystem and includes exceeding its specified allowable degradation of essential subsystem functions.

3.2.4 Maintainability

3.2.4.1 Mean Time to Repair Critical Functions (MTTRCF)

PAVE PILLAR system shall have a mean time to repair critical functions of 1.25 hours. The mean time to repair a critical function includes 1) turn on and stabilization, 2) fault verification, 3) on-equipment troubleshooting time, 4) repair, 5) functional checkout, 6) turn off and secure.

3.2.4.2 On-Equipment Troubleshooting Time

The goal for on-equipment troubleshooting time is 10 minutes or less. On-equipment troubleshooting time is the

average time required to isolate malfunctions using BIT support equipment and/or tech orders to the lowest level indenture where corrective maintenance can be performed.

3.2.4.3 Percent Fault Detection

The goal for the PAVE PILLAR system is 99% fault detection. The percent fault detection is determined by the following computation. The number of verified failures detected by BIT divided by the total number of verified failures detected by all other methods. A verified failure is a condition where 1) equipment performance, including bit performance, is less than that required by specifications, and 2) corrective action is required to restore equipment performance. The fault detection percentage applies to all possible faults as weighted by the relative rates. This format assumes that a requirement already exists for 100% diagnostic capability.

3.2.4.4 Percent Fault Isolation

The goal for the PAVE PILLAR avionics is 98% fault isolation. Percent fault isolation is computed by the following method. The number of malfunctions properly isolated by bit divided by the number of verified failures detected by bit. Proper isolation is defined as unambiguous identification of a single unit that is of the lowest level of assembly that is defined for removal from the end item.

3.2.5 Availability

3.2.5.1 Sortie Rate

The goal for sortie rate per day is 4.5 or greater.

3.2.5.2 Abort Rate

The goal for abort rate shall be 1% or less.

3.2.5.3 Combat Turnaround Time

The PAVE PILLAR design will impose no system requirements which would preclude a combat turnaround time of 15 minutes. Combat turnaround time is defined as the elapsed time required to inspect, clean, service fuel, oil, hydraulics, and other consummables, re-arm, load, and prepare to relaunch an aircraft returning in a mission capable status. This time may also include a re-key or update to the COMSEC/TRANSEC systems. Time begins when the aircraft stops taxiing and ends when the aircraft begins taxiing toward the runway for a subsequent sortie.

3.2.5.4 Total Non-Mission Capable for Maintenance of Avionics

The total non-mission capable for maintenance of avionics shall not exceed 1.2%. Total non-mission capable for maintenance of avionics is the cumulative percentage (based on 24 hour clock) of aircraft that are non-mission capable because of avionics maintenance.

3.2.5.5 Total Non-Mission Capable for Supply of Avionics

Total non-mission capable for supply of avionics shall not exceed 0.2%. Total non-mission capable for supply of avionics is the cumulative percentage based on the 24 hour clock of aircraft that are non-mission capable because they need avionics parts.

3.2.6 Effectiveness Models (Not Applicable)

3.2.7 Environmental Conditions (Not Applicable)

3.2.8 Nuclear Control (Not Applicable)

3.2.9 Transportability (Not Applicable)

3.3 Design and Construction

3.3.1 Material, Parts and Processes

The PAVE PILLAR Avionics shall be designed to take maximum advantage of VHSIC and fiber optic technology.

3.3.2 Electromagnetic Compatibility (Not Applicable)

3.3.3 Nameplate and Product Marking (Not Applicable)

3.3.4 Workmanship (Not Applicable)

3.3.5 Interchangeability

The Common Modules which make up the PAVE PILLAR Avionics (see Section 6.1) will be interchangeable within type classes, i.e., VHSIC 1750A Processors, Avionics Bus Interface, etc.

3.3.6 Safety (Not Applicable)

3.3.7 Human Engineering

All man-machine interactions shall be designed so as to provide maximum off-loading of tasks which can be accomplished by a computer system and to place minimum interpretive demands on the crew. The crew should be presented with clear, concise, easily interpreted display data, prompts, selection ranges, etc. and a minimum of human intervention in mission

control functions should be required. Nevertheless, mission-authorized manual control/override of automated avionics functions shall be made possible at all times; the responsibility for any reconfiguration or reparameterization required to support manual operations rests with the Mission Supervisor.

3.3.8 Computer Program Design

3.3.8.1 Higher Order Language/Run Time Support System

The Operational Flight Program (OFP) will be developed in accordance with the requirements of the Ada (ANSI/MIL-STD-1815A) programming language, unless otherwise approved by the Government.

3.3.8.2 Software Development System/Design Aids

Software development for operational flight programs will be accomplished using an integrated Ada Programming Support Environment (APSE) which supports a standard set of programming tools (editor, compiler, linker, loader, debugger) in addition to an Ada based program design language (PDL) for use in the design stage of the software development. The software development system will support the MIL-STD-1750A data processor instruction set as a target machine.

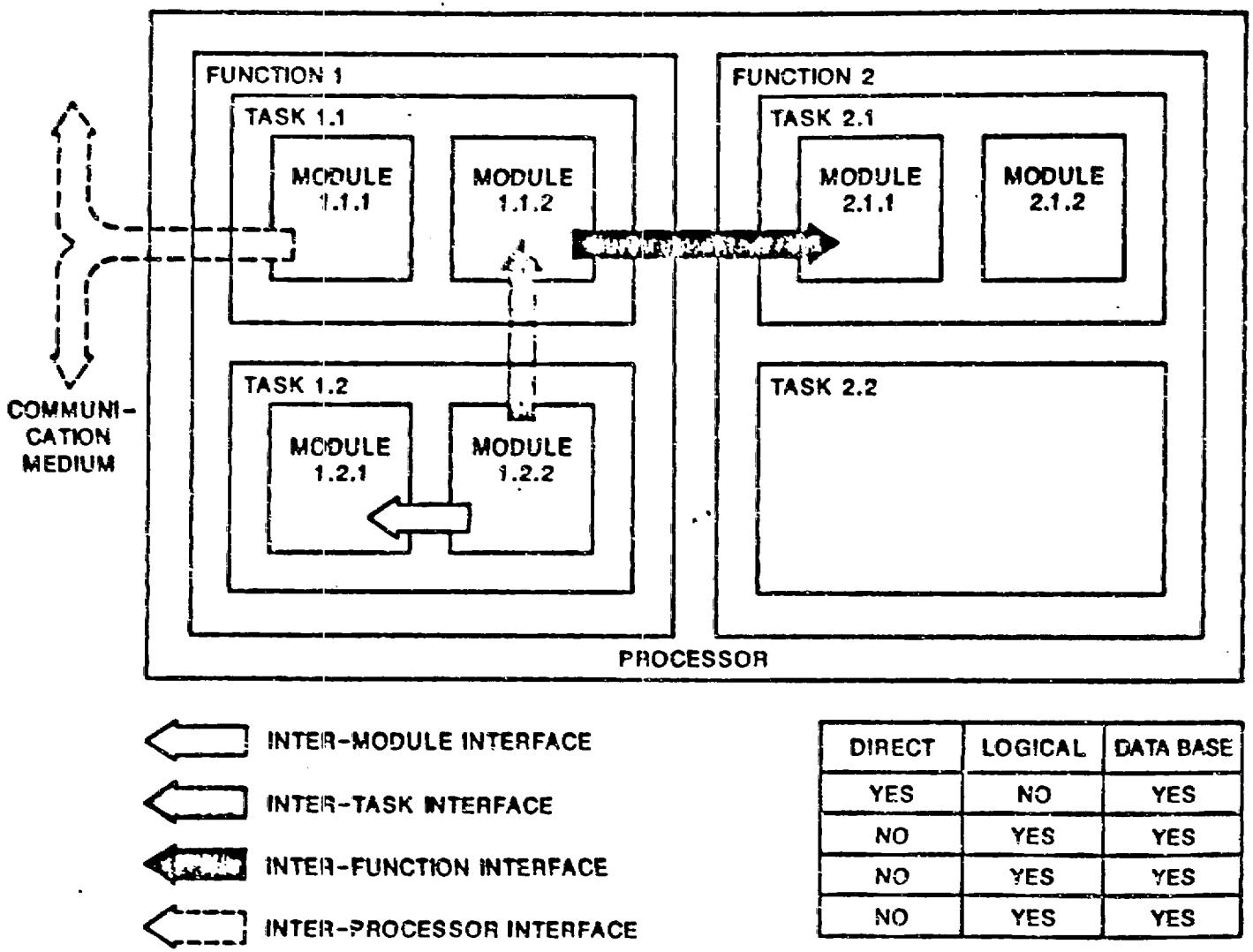
3.3.8.3 Standard Interfaces Requirement

All PAVE PILLAR application software will observe rigid interfaces between tasks. Interfaces between functions will also observe inter-task interface restrictions, and will utilize such interfaces only as are consistent with functional isolation between functions. The trade-off between functional isolation and the need for integration functions to satisfy PAVE PILLAR reliability requirements is simplified by the application of rigidly enforced interface definitions.

Four general classes of interface will exist: interfaces between one executional level and another, as typified by a subroutine call, which exist within a task; interfaces between two tasks which constitute some portion of a major function; interfaces between two functions; and interfaces between software executing in two or more processors. The last class of interfaces can coexist with any of the other three, and will typically coexist with inter-function interfaces. These classes are illustrated in Figure 3.3.8.3-1.

The least restricted class of interfaces is the class of inter-module interfaces. These interfaces must not transcend task boundaries, or they become inter-task interfaces. The general rule for inter-module interfaces are defined by the programming language, or by convention for assembly language modules. The application software will be written in accordance with well defined inter-module interfaces, which will be specified to enhance modularity by maximum suitable use of parametric and table-driven techniques.

Inter-task interfaces are more restrictive than inter-module interfaces. In this context, the tasks are part of the same function, since an interface between tasks of two different functions are inter-function interfaces. Inter-task interfaces are not permitted to be directly executed, but instead require executable intervention. Consequently, this class of interfaces is restricted to the use of logical events and data base accesses. The exception to this is the interface with the Kernel Executive, which for this purpose is considered to be a part of every task. Thus, a task can call the executive using inter-module techniques, and the executive can perform the requested operation on another task, also using inter-module techniques since the executive is a part of both tasks. A task thus affects the operation of another task by a logical operation. The interfaces between the tasks and the



DIRECT	LOGICAL	DATA BASE
YES	NO	YES
NO	YES	YES
NO	YES	YES
NO	YES	YES

Figure 3.3.8.3-1 CLASSES OF INTERFACES

executive are, therefore, required to be well defined, rigidly applied, and standardized for all tasks. These interfaces are further discussed in Section 6.3.

The class of inter-function interfaces is more restrictive than inter-task interfaces. The interfaces represent a trade-off between the need for functional integration. Functional isolation, at the ideal, demands a total separation of functions. The ideal for integration demands use the maximum amount of interchange between functions to permit liberal transfer of information. The use of data base information as the transfer vehicle does not seriously denigrate isolation while greatly enhancing integration from the fully isolated condition. Likewise, logical effects can be useful as long as the vehicle for transfer of the effect is structured to provide the maximum amount of isolation. Direct interface defeats isolation entirely. It, therefore, appears that the inter-function/inter-task classes should be minimized for the former and freely permitted for the latter.

Inter-processor interfaces can occur for any other class of interface. It is highly inefficient for an inter-module interface to be implemented between processors due to the overhead associated with coordinating control and data base transfer between processors. This will, therefore, not be permitted for application software. As discussed earlier, to permit inter-function interfaces is functionally to provide the capability for inter-task interfaces. Inter-function interface capability is required in an inter-processor environment. It is recommended that inter-function/inter-processor interfaces be restricted by convention to minimize the amount of data necessary to be transferred in support of the interface. The capability will be provided, with the limitation being conventional and subject to permissable partitioning details. All inter-processor interfaces will be effected by intervention

of both the Kernel and Distributed Executives of the respective processors.

3.4 Documentation

Elements which require additional documentation include as a minimum the 1750 Processor, Common Signal Processor, High Speed Data Bus, Switch Assemblies, and other major architecture elements.

3.5 Logistics

The Avionics System shall be designed with a primary goal of minimizing the costs of operation and maintenance and to enhance the accomplishment of required maintenance.

3.5.1 Maintenance Environment

The Avionics System will be designed to support deployment to sites with damaged, limited or non-existent support facilities.

3.5.1.1 Maintenance Concept

The PAVE PILLAR Maintenance Concept is based upon two levels of maintenance: On-Equipment (aircraft) and Off-Equipment (depot).

3.5.1.1.1 Organizational Level Maintenance Requirements

A. Maintenance shall consist of repair by replacement of faulty/failed items.

1) On-equipment repairs shall be performed on miscellaneous wiring, cabling, and connectors.

B. Packaging and modularization of system items shall be designed to ensure that maintenance personnel are able to remove and return faulty/failed items to depot repair facilities.

C. The primary fault detection and isolation (FDI) functions for corrective maintenance and system readiness tests shall be provided by a combination of on-board diagnostics and built-in-test (BIT).

1) Provisions for manual testing with in-place common test equipment shall be provided when required.

2) On-board diagnostics and BIT shall include automated maintenance aids to direct maintenance personnel to faulty/failed items with clear, concise statements describing the item, the type of fault or failure, and sequence of events in time, and the location of the item with text and graphic line drawings when required.

3) On-board diagnostics and BIT results shall be recorded in a non-volatile medium for recall capability inflight or postflight and transfer capability to ground maintenance facilities.

D. No peculiar support equipment shall be required to complete organization level maintenance.

E. No preventive (scheduled) maintenance shall be required except system readiness tests.

3.5.1.1.2 Depot Level Maintenance Requirements

A. Depot level maintenance facilities shall be responsible for the following actions:

1) Repair the items forwarded from organizational facilities and return them to the supply system.

2) Condemn the item and notify supply of the disposition.

3.5.2 Supply Support

Supply Support will be tailored to Military Supply and Transportation Evaluation Procedures (MILSTEP). The Organizational Maintenance Unit will be authorized Spare Parts based upon the two level maintenance concept. Supply levels will be authorized based upon usage factors and pipeline times plus a safety level. A supply of Line Replaceable Modules will be authorized.

3.5.3 Facilities (Not Applicable)

3.5.4 Cost

The PAVE PILLAR concept requires an overall lower life cycle cost with respect to current avionics suites. The required characteristics include:

o Use of developing technology applications, i.e., VHSIC.

o Standardization of data processors (MIL-STD-1750A), Multiplex Bus Communications (MIL-STD-1553B and High Speed Data Bus), and operating system and application software (MIL-STD-1815A).

o Modularization and replication of modules across functions, as well as to achieve redundancy within a function. Modularization is carried to the lowest possible

level to facilitate remove and replace maintenance in a two-level maintenance concept.

- o High reliability specifications for critical system components.

- o Fault detection and isolation through on-board diagnostics and built-in-test (BIT).

- o Ease of Pre-Planned Product Improvement (P^3I).

3.6 Personnel and Training (Not Applicable)

3.7 Functional Area Characteristics

The PAVE PILLAR core avionics is partitioned into four functional groups:

(1) Digital Signal Processing
(2) Mission Processing
(3) Vehicle Management System
(4) Avionics System Control.

To gain maximum commonality across functional areas and to minimize the number of interface module types, all system busses as a class and all system data distribution networks as a class should be of a common functional type. All communication paths between system elements should employ fiber optic media.

The common components and their internal interfaces are described in the following subsections: Data Processors, System Mass Memory, Mission Avionics Bus, Block Transfer Bus, Vehicle Management System Bus, Signal Processor, Sensor Data Distribution Network, Video Data Distribution Network, System

Executive, Kernel Executive and Distributed Executive. See Section 6.1 for a description of individual modules.

3.7.1 Data Processors

The Data Processors perform the general purpose computational processing that does not require unique I/O interfaces, dedicated hardware functions or excessive throughput rate. Each processor will conform to the MIL-STD-1750A Instruction Set Architecture (ISA). Each data processor will contain one or more V1750A processor modules on the Common Backplane Communication Bus.

3.7.1.1 Data Processor Diagrams

The Data Processsor for the Mission Processing Area and Vehicle Management Area are shown in Figs. 3.7.1-1 and 3.7.1-2.

3.7.1.2 Operating System

Each Data Processor's Operating System will be comprised of a Distributed Executive and a Kernel Executive.

3.7.1.3 Inter-Module Communications

The interface between modules shall be provided by the integrated rack. The integrated rack shall provide a common backplane which shall interconnect the modules.

3.7.1.3.1 Inter-Module Communications Bus (PI-Bus)

The PI-Bus shall be the primary general purpose parallel backplane bus which interconnects sets of modules performing a common function (i.e., Data Processor). The PI-Bus Pave Pillar implementation shall be 16 data bits with

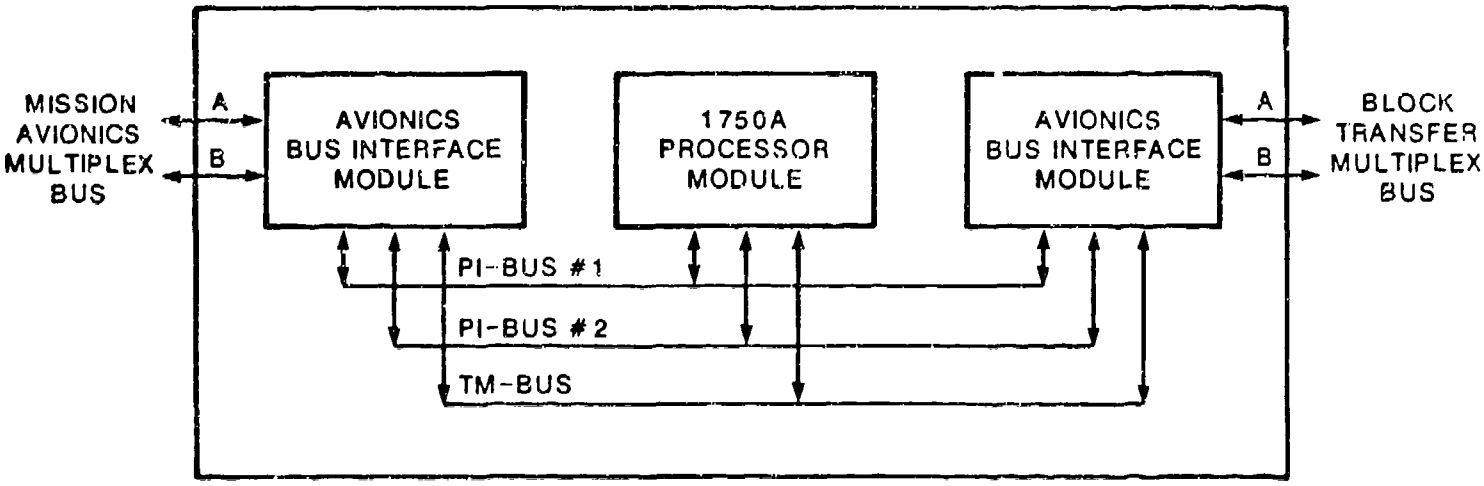


Figure 3.7.1-1 DATA PROCESSOR MODULAR STRUCTURE

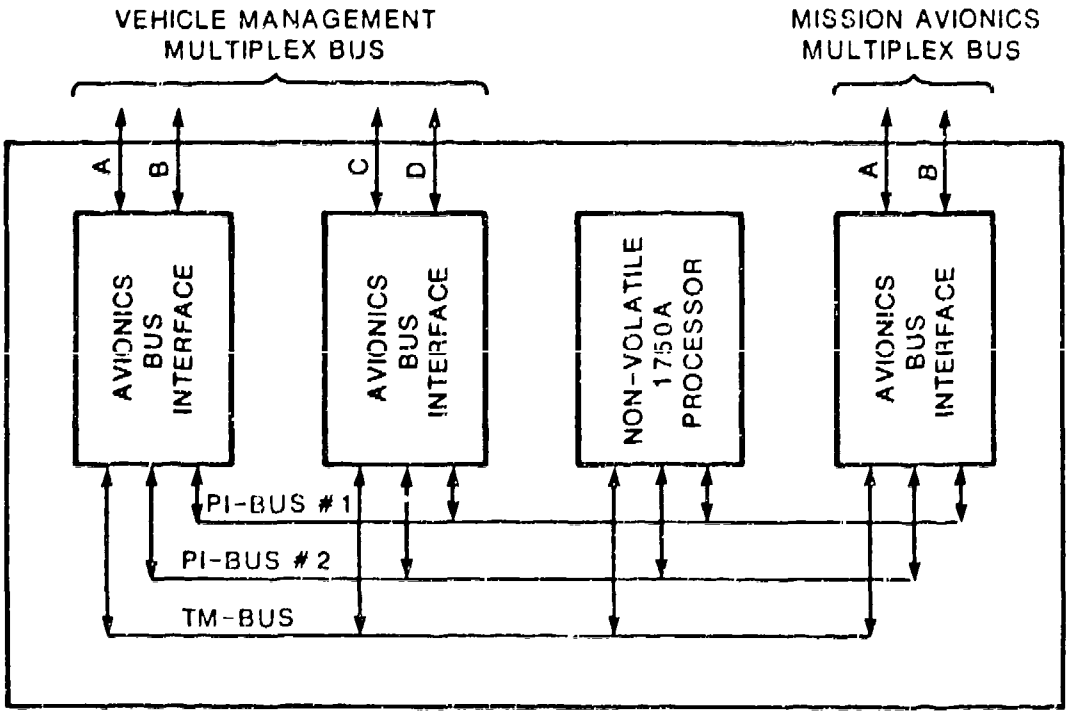


Figure 3.7.1-2 VEHICLE MANAGEMENT SYSTEM DATA PROCESSOR MODULAR STRUCTURE

error detection. (See Appendix B - VHSIC Phase 2 Interoperability Standards, PI-Bus Specification) The PI-Bus shall support the transfer of 12.5 million words per second. Two PI-Buses shall be employed for redundancy in case of primary bus failure.

3.7.1.3.2 Test and Maintenance Bus (TM-Bus)

The Test and Maintenance Bus shall be a serial 4-signal backplane bus which interconnects sets of modules performing a common function (See Appendix C - VHSIC Phase 2 Interoperability Standards TM-Bus Specification). The primary purpose of the TM-Bus is to provide a common means for performing test and maintenance of modules at the circuit level at the depot or factory. It is also intended to be used to support on-equipment maintenance to detect and isolate faults to a module. The TM-Bus may be used operationally to perform fault detection and isolation on those modules designed to use the TM-Bus in that manner. It shall allow the module designated as a maintenance controller (the VHSIC 1750A CPU module shall have maintenance controller capability) to initiate/command self test functions on all of the modules, as well as support the collection of self test status information and make that status available external to the functional area.

3.7.2 System Mass Memory

The System Mass Memory shall be non-volatile and have both read/write capabilities. The System Mass Memory shall be able to act as a logical file oriented device with an intelligent controller interface as well as a random access storage device.

3.7.2.1 System Mass Memory Diagram

The System Mass Memory Diagram is shown in Figure 3.7.2.1-1.

3.7.2.2 Inter-Module Communications

The interface between modules shall be provided by the integrated rack. The integrated rack shall provide a common back plane which shall interconnect the modules.

3.7.2.2.1 Inter-Module Communications Bus (PI-Bus)

The PI-Bus shall be the primary general purpose parallel backplane bus which interconnects sets of modules performing a common function (i.e., Data Processor). The PI-Bus Pave Pillar implementation shall be 16 data bits with error detection. (See Appendix B - VHSIC Phase 2 Interoperability Standards, PI-Bus Specification.) The PI-Bus shall support the transfer of 12.5 million words per second. Two PI-Buses shall be employed for redundancy in case of primary bus failures.

3.7.2.2.2 Test and Maintenance Bus (TM-Bus)

The Test and Maintenance Bus shall be a serial 4-signal backplane bus which interconnects sets of modules performing a common function (See Appendix C - VHSIC Phase 2 Interoperability Standards TM-Bus Specification). The primary purpose of the TM-Bus is to provide a common means for performing test and maintenance of modules at the circuit level at the depot or factory. It is also intended to be used to support on-equipment maintenance to detect and isolate faults to a module. The TM-Bus may be used operationally to perform fault detection and isolation on those modules designed to use the TM-Bus in that manner. It shall allow the module

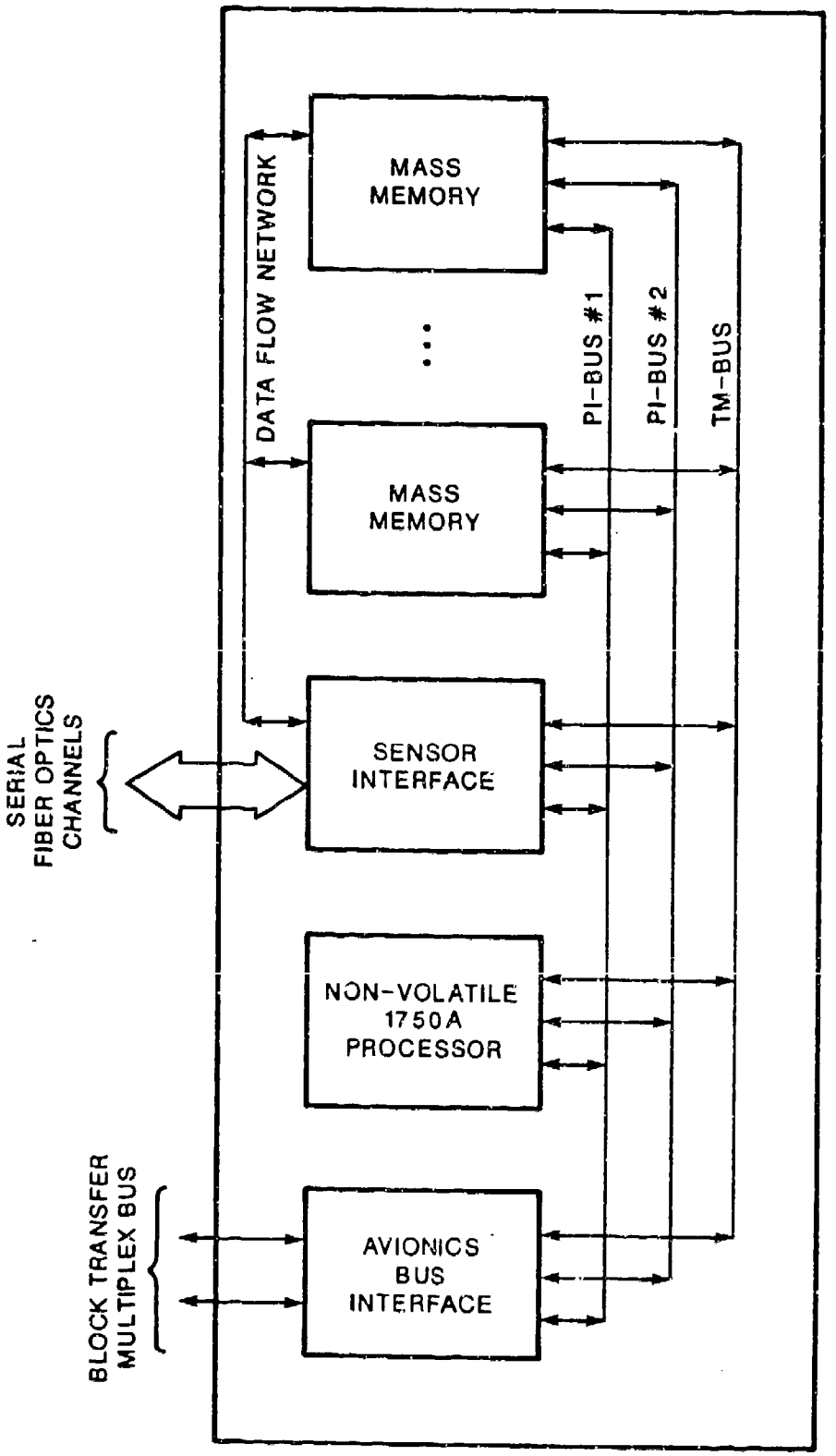


Figure 3.7.2.1-1 SYSTEM MASS MEMORY MODULAR STRUCTURE

designated as a maintenance controller (the VHSIC 1750A CPU module shall have maintenance controller capability) to initiate/command self test functions on all of the modules, as well as support the collection of self test status information and make that status available external to the functional area.

3.7.3 Universal Remote Terminals

The Universal Remote Terminals shall provide: (1) interfaces to non-core devices such as data recorders, data transport units and helmet voice/sight units, (2) a method of interfacing standard test equipment to the on-board computer systems using IEEE 488 standard test equipment buses, and (3) a MIL-STD-1553B interface capability.

3.7.3.1 System Diagram

A typical Universal Remote terminal is shown in Figure 3.7.3-1.

3.7.3.2 Inter-Module Communications

The interface between modules shall be provided by the integrated rack. The integrated rack shall provide a common back plane which shall interconnect the modules.

3.7.3.2.1 Inter-Module Communications Bus (PI-Bus)

The PI-Bus shall be the primary general purpose parallel backplane bus which interconnects sets of modules performing a common function (i.e., Data Processor). The PI-Bus Pave Pillar implementation shall be 16 data bits with error detection. (See Appendix B - VHSIC Phase 2 Interoperability Standards, PI-Bus Specification.) The PI-Bus shall support the transfer of 12.5 million words per second.

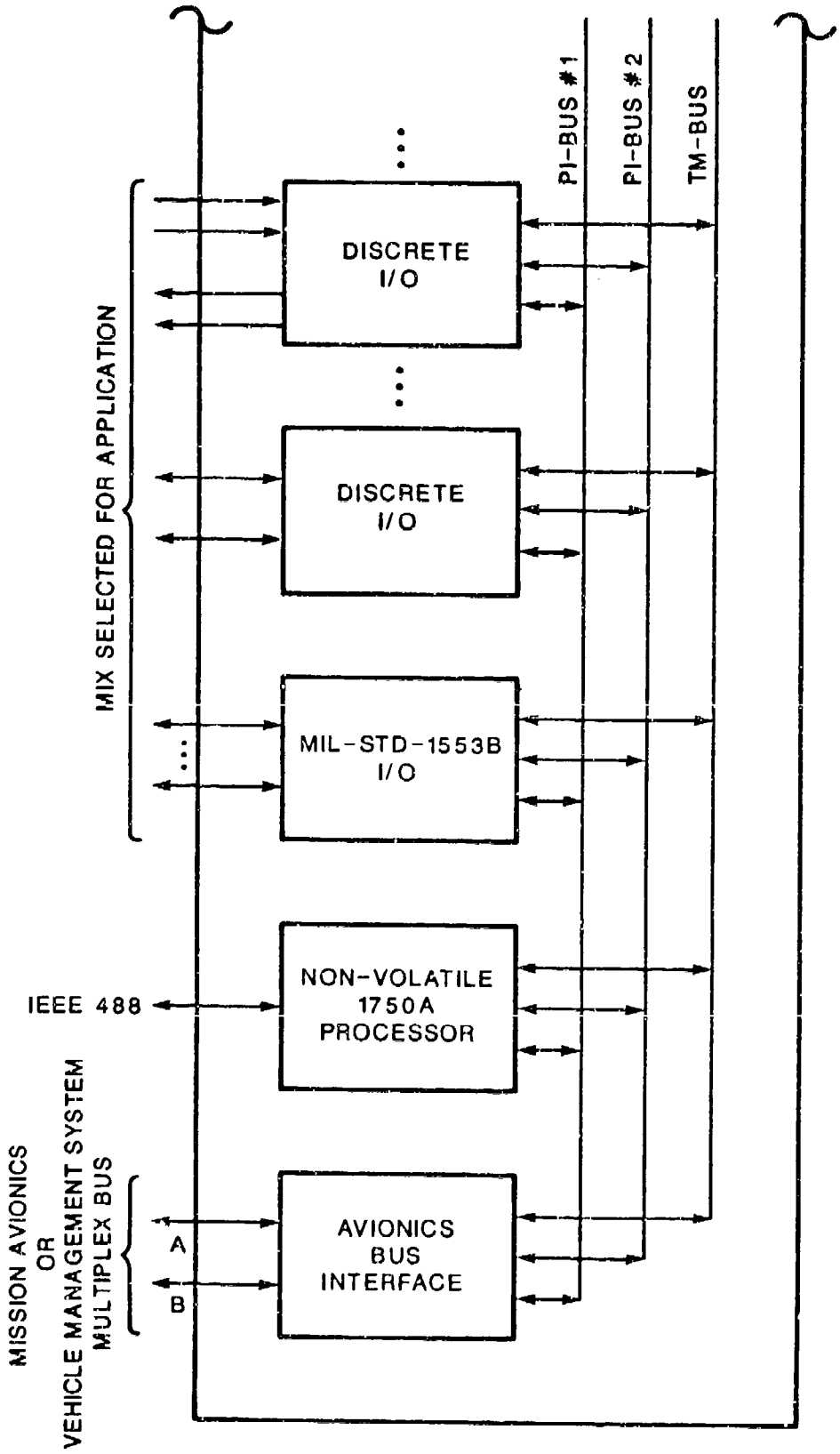


Figure 3.7.3-1 UNIVERSAL REMOTE TERMINAL MODULE STRUCTURE

Two PI-Buses shall be employed for redundancy in case of primary bus failure.

3.7.3.2.2 Test and Maintenance Bus (TM-Bus)

The Test and Maintenance Bus shall be a serial 4-signal backplane bus which interconnects sets of modules performing a common function (See Appendix C - VHSIC Phase Interoperability Standards TM-Bus Specification. The primary purpose of the TM-Bus is to provide a common means for performing test and maintenance of modules at the curcuit level at the depot or factory. It is also intended to be used to support on-equipment maintenance to detect and isolate faults to a module. The TM-Bus may be used operationally to perform fault detection and isolation on those modules designed to use the TM-Bus in that manner. It shall allow the module designated as a maintenance controller (the VHSIC 1750A CPU module shall have maintenance controller capability) to initiate/command self test functions on all of the modules, as well as support the collection of self test status information and make that status available external to the functional area.

3.7.4 Mission Avionics Multiplex Bus

The Mission Avionics Multiplex Bus shall consist of two independent/redundant buses. Each bus will have a minimum effective data transmission rate of 20 MBPS.

3.7.4.1 Topology

The Mission Avionics Multiplex Bus shall use a logical multidrop linear bus structure. The implementation shall be capable of supporting 64 physically separate remote terminals.

3.7.4.2 Protocol

The Mission Avionics Multiplex Bus shall provide the following protocol:

- Physical (Terminal) and logical (content) Addressing Modes
- Synchronous and Asynchronous Messages
- Bounded latencies
- Lost message determination
- Terminal Insertion and removal
- System Initialization

(See Appendix D - High Speed Data Bus System Specification).

3.7.4.3 Control

The Mission Avionics Multiplex Bus shall implement a distributed control mechanism. The control mechanism shall support asynchronous and priority messages. The control mechanism shall be deterministic.

3.7.4.4 Interface Characteristic

Each unit connected to the Mission Avionics Multiplex Bus shall provide an Avionics Bus Interface Module.

3.7.4.5 Physical Characteristics

The preferred media for the Mission Avionics Multiplex Bus shall be fiber optics with a passive, transmissive star coupler.

3.7.4.6 Fault Recovery

All fault recovery such as lost tokens, terminal drop out, lost media, lost message, etc., shall recover within

20 milliseconds. Faults requiring more recovery time shall be deemed not recoverable.

3.7.4.7 Nuclear Hardness

The VHSIC Phase I nuclear hardness requirements should be used.

3.7.5 Block Transfer Bus

The Block Transfer Bus shall consist of two independent/redundant buses. Each bus will have a minimum effective data transmission rate of 20 MBPS.

3.7.5.1 Topology

The Block Transfer Bus shall use a logical multidrop linear bus structure. The implementation shall be capable of supporting 64 physically separate remote terminals.

3.7.5.2 Protocol

The Block Transfer Bus shall provide the following protocol:

- Physical (Terminal) and logical (content) Addressing Modes
- Synchronous and Asynchronous Messages
- Bounded latencies
- Lost message determination
- Terminal Insertion and removal
- System initialization

(See Appendix D - High Speed Data Bus System Specification).

3.7.5.3 Control

The Block Transfer Bus shall implement a distributed control mechanism. The control mechanism shall support asynchronous and priority messages. The control mechanism shall be deterministic.

3.7.5.4 Interface Characteristics

Each unit connected to the Block Transfer Bus shall provide an Avionics Bus Interface Module.

3.7.5.5 Physical Characteristics

The preferred media for the Block Transfer Bus shall be fiber optic with a passive, transmissive star coupler.

3.7.5.6 Fault Recovery

All fault recovery such as lost tokens, terminal drop out, lost media, lost message, etc., shall recover within 20 milliseconds. Faults requiring more recovery time shall be deemed not recoverable.

3.7.5.7 Nuclear Hardness

The VHSIC Phase I nuclear hardness requirements should be used.

3.7.6 Vehicle Management System Multiplex Bus

The Vehicle Management System Multiplex Bus shall support four independent buses. Each bus will have a minimum effective data transmission rate of 20 MBPS.

3.7.6.1 Topology

The Vehicle Management System Multiplex Bus shall use a logical multidrop linear bus structure. The implementation shall be capable of supporting 64 physical separate remote terminals.

3.7.6.2 Protocol

The Vehicle Management System Multiplex Bus shall provide the following protocol:

- Physical (Terminal) and logical (content) Addressing Modes
- Synchronous and Asynchronous Messages
- Bounded latencies
- Lost message determination
- Terminal Insertion and removal
- System Initialization

Appendix D - High Speed Data Bus Protocol describes the bus protocol.

3.7.6.3 Control

The Vehicle Management System Multiplex Bus shall implement a distributed Control Mechanism. The Control Mechanism shall support asynchronous and priority messages. The control mechanism shall be deterministic.

3.7.6.4 Interface Characteristics

Each unit connected to the Vehicle Management Multiplex Bus shall provide two Avionic Bus Interface Modules.

3.7.6.5 Physical Characteristics

The preferred media for the Vehicle Management System Multiple Bus shall be fiber optic transmission media with a passive, transmissive star coupler.

3.7.6.6 Fault Recovery

All fault recovery such as lost tokens, terminal drop out, lost media, lost message, etc. shall recover within 20 milliseconds. Faults requiring more recovery time shall be deemed not recoverable.

3.7.6.7 Nuclear Hardness

The VHSIC Phase I nuclear hardness requirements should be used.

3.7.7 Common Signal Processor (CSP)

The Common Signal Processor consists of modular resources that perform the signal processing function for Radar, EW, Image and CNI Processing. Table 3.7.7-1 lists typical signal processing algorithms the CSP shall be capable of implementing in real time.

3.7.7.1 Common Signal Processor Diagram

The Common Signal Processor is shown in Figure 3.7.7.1-1.

3.7.7.2 Inter-Module Communications

The interface between modules shall be provided by the integrated rack. The integrated rack shall provide a common back plane which shall integrate the modules.

TABLE 3.7.7-1
TYPICAL SIGNAL PROCESSING FUNCTIONS

RADAR	EW
FFT, FFT^{-1}	Table Lookup
Coordinate Transformation	Polynomial Evaluation
Complex Multiply	Chinese Remainder
Complex Aid	Associative Memory
Floating-Point Multiply/Add	Hashing
Lookup Multiply	Decision Trees
Normalize Divide	Window Compares
Averaging	Multiparameter Correlation
Thresholding	Statistical Filter (e.g., Kalman)
Vector/Matrix Operations	Dynamic Memory Assignment
Matrix Transpose	Demodulation Algorithms
Chinese Remainder	Statistics (Mean, Standard Deviation)
Logarithm	
Canceller	
Compare	**IMAGE PROCESSING**
Max/Min	
Accumulate	Adaptive Lattice Filter
Trig Functions (Weighting)	Discrete Convolution
Sliding Window	Recursive Filtering
Convolution	Discrete Cosine Transform
FIR Filter	Walsh-Hadamard Transform
IIR Filter	Haar Transform
	Gain and Bias Correction
	Direct Correlation
	X-Form Correlation
	Gradient Operator
	Kirsch Edge Filter
	Sobel Edge Filter
	Cubic Convolution Resampling
	Histogram Modification

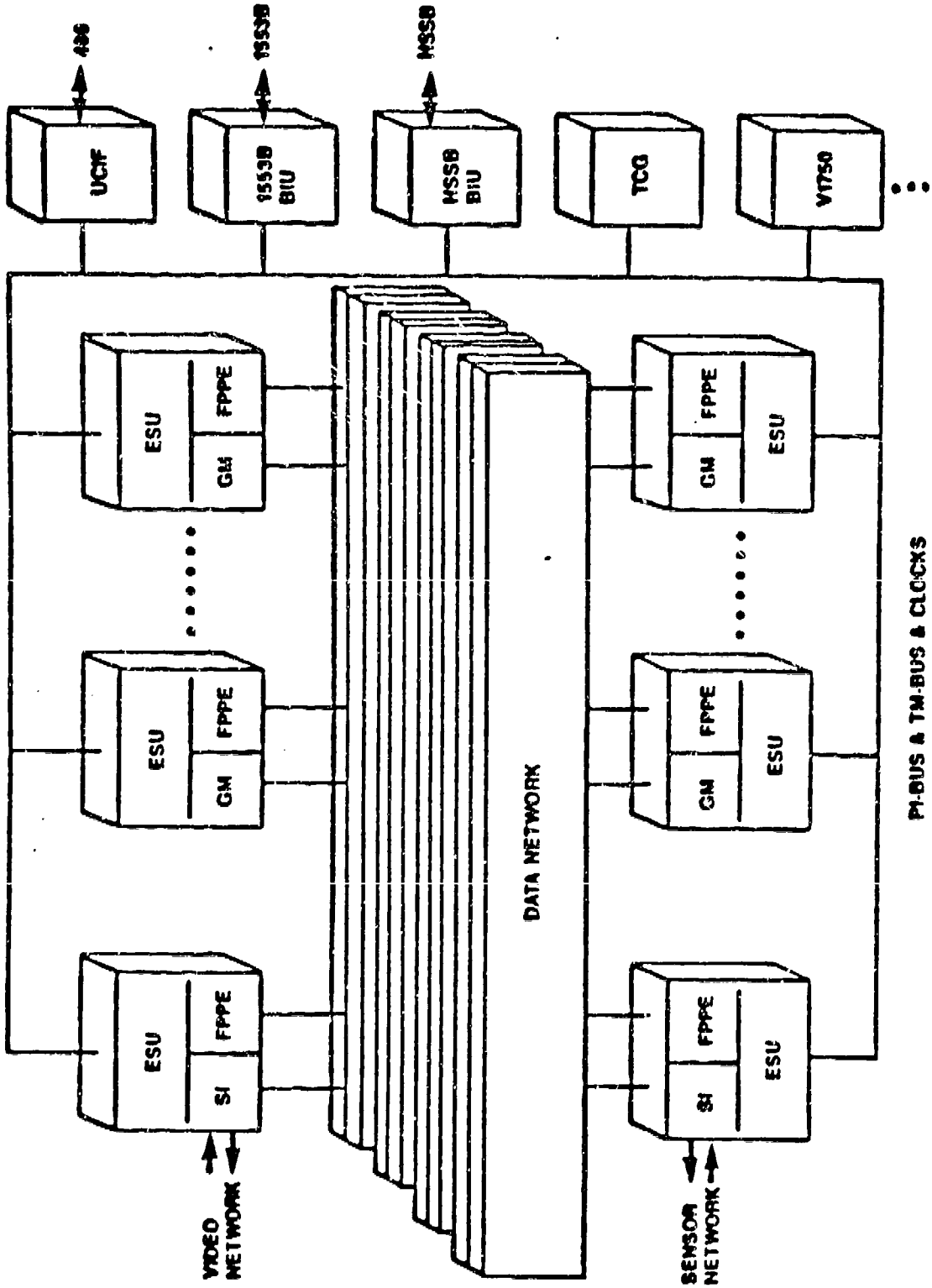


Figure 3.7.7.1-1 COMMON SIGNAL PROCESSOR HARDWARE ARCHITECTURE

3.7.7.2.1 Inter-Module Communications Bus (PI-Bus)

The PI-Bus shall be the primary general purpose parallel backplane bus which interconnects sets of modules performing a common function (i.e., Data Processor). The PI-Bus Pave Pillar implementation shall be 16 data bits with error detection. (See Appendix B - VHSIC Phase 2 Interoperability Standards, PI-Bus Specification.) The PI-Bus shall support the transfer of 12.5 million words per second. Two PI-Buses shall be employed for redundancy in case of primary bus failure.

3.7.7.2.2 Data Flow Network

The internal Data Flow Network provides high speed data transmission paths between modules performing a common function (i.e., signal processing). The Data Flow Network shall be a network switching concept which supports simultaneous interconnections of any independent pair of sources and destinations of data connected to the network. Control of the network is decentralized: the initiating module determines the path through the network and the target module for the transmission. The network is optimized for block transfers; it may be implemented with a variety of path widths, including 16-bit and 32-bit data paths. The baseline word width is 32 bits. The transfer rate goal is 25 million words per second. The networks may be implemented with a variety of numbers of ports; the maximum number shall be at least 24 ports, supporting up to 12 parallel transmissions. Any port may be connected to any other port.

3.7.7.2.3 Test and Maintenance Bus (TM-Bus)

The Test and Maintenance Bus shall be a serial 4-signal backplane bus which interconnects sets of modules performing a common function (See Appendix C - VHSIC Phase 2

Interoperability Standards TM-Bus Specification. The primary purpose of the TM-Bus is to provide a common means for performing test and maintenance of modules at the circuit level at the depot or factory. It is also intended to be used to support on-equipment maintenance to detect and isolate faults to a module. The TM-Bus may be used operationally to perform fault detection and isolation on those modules designed to use the TM-Bus in that manner. It shall allow the module designated as a maintenance controller (the VHSIC 1750A CPU module shall have maintenance controller capability) to initiate/command self test functions on all of the modules, as well as support the collection of self test status information and make that status available external to the functional area.

3.7.8 Sensor Data Distribution Network

The Sensor Data Distribution Network provides the Signal Processor with conditioned sensor/preprocessor input. The SDDN allows any sensor to directly connect to any Signal Processor with a unidirectional data flow of 500 MBPS per channel. The SDDN consists of fiber optic connections from each sensor front-end to a routing network and then to each Signal Processor. The routing network consists of matrix switch modules that provide simultaneous non-blocking connection of multiple fiber optic channel ports. The SDDN has redundant channels and is reconfigured only upon the occurrences of functional Signal Processing Area failures or upon major mission mode changes.

A portion of the SDDN, the Sensor Control Network, provides the communication path from the Signal Processors back to the Sensor Subsystems. Moding, control data and signal data flow from Signal Processors to a routing network and then to sensor front-ends via unidirectional 500 MBPS fiber optic channels.

3.7.8.1 System Diagram

A diagram of the network is shown in Figure 3.7.8-1.

3.7.9 Data Exchange Network

The Data Exchange Network shall permit the loading of Signal Processing area memories, the exchange of data between Signal Processors, the routing of encrypted data (black) from TRANSEC/COMSEC Secure Data Units (SDUs) to the Signal Processing area and the routing of data (red) from the Signal Processing area to the TRANSEC/COMSEC SDUs. The Data Exchange Network will permit a unidirectional flow of 500 MBPS per channel. The Data Exchange Network shall use the same network approach common to the SDDN and the VDDN.

3.7.9.1 System Diagram

A diagram of the network is shown in Figure 3.7.9-1.

3.7.10 Video Data Distribution Network

The Video Data Distribution Network provides the connectivity between all video data sources and sinks. Video sources are the Signal Processors and Stores Management System; video data sinks are the cockpit displays and mission devices such as a Video Tape Recorder. The VDDN transfers digital graphics data in a format common to all source and sink devices. The VDDN routing network is functionally equivalent to the SDDN and DEN using common fiber optic channels, switch implementations, and interface modules. The VDDN shall support six simultaneously active display channels. A minimum of four 525 line resolution and two 875 line resolution color pictures at a 30 HZ update rate shall determine the minimum composite transmission rate of the VDDN.

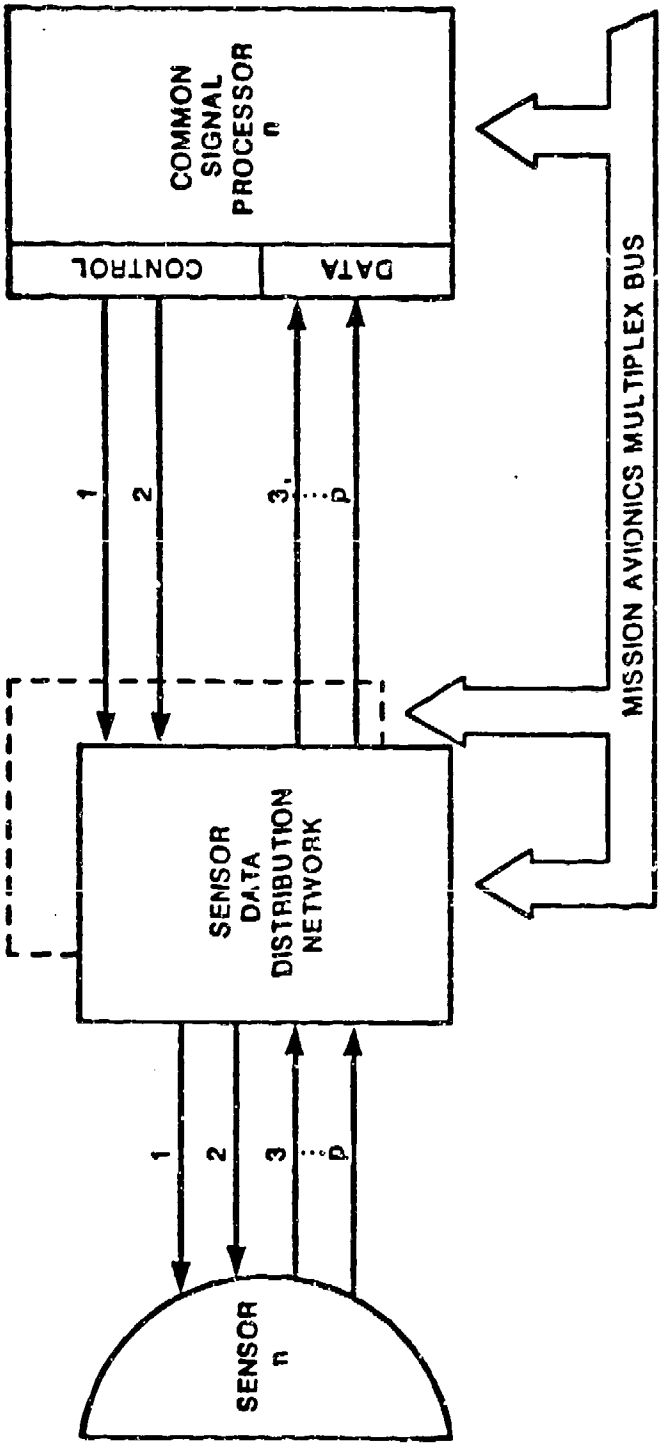


Figure 3.7.6-1 SENSOR DATA DISTRIBUTION NETWORK

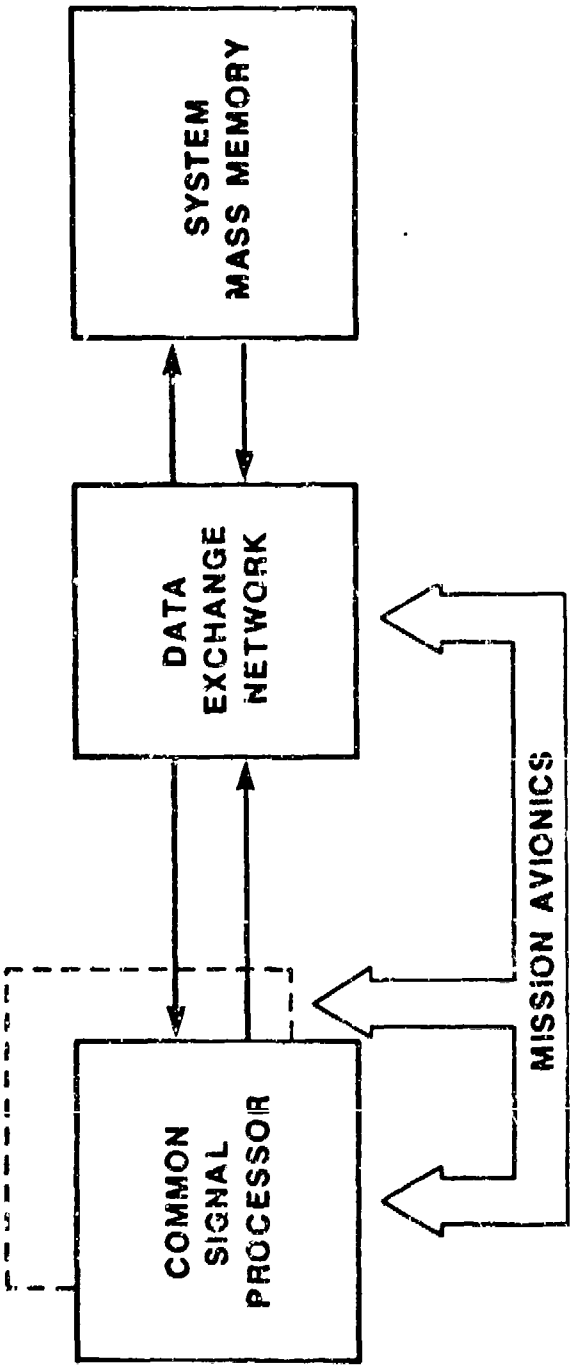


Figure 3.7.9-1 DATA EXCHANGE NETWORK

3.7.10.1 System Diagram

A diagram of the network is shown in Figure 3.7.10-1.

3.7.11 System Executive

The System Executive resides in the Mission Data Processors. There shall be two copies, one of which is designated the primary (System Supervisor), and the other is designated the standby. The standby system will operate in a "hot backup" mode, available to take over whenever a fault occurs in the primary. The System Executive is responsible for monitoring the state of the system hardware and software. The primary means of system status monitoring, communication, and logging will be a system data base which is located in each of the data processors and is maintained by the Distributed Executive. Changes of device and task status will be broadcast over the high speed data bus as an update to the local copies of the state tables held in each processor and in the System Mass Memory. The System Executive is responsible for keeping a fault log for all system components and for controlling reconfiguration of the system resources. The System Executive handles two types of system resource reallocation. First, as functional requirements change throughout the mission, the system components must be reallocated to meet these requirements. These requirements represent the functional capabilities which were defined during the mission planning process and modified by any subsequent inputs. Second, when any system component currently in use is determined to have failed, reallocation of the remaining components may be necessary in order to retain any functionality lost by the failure. If the mission proceeds as planned, the pilot will be provided with the expanded functional capabilities and the System Executive resource allocation process will be transparent. Whenever any component is lost, through internal

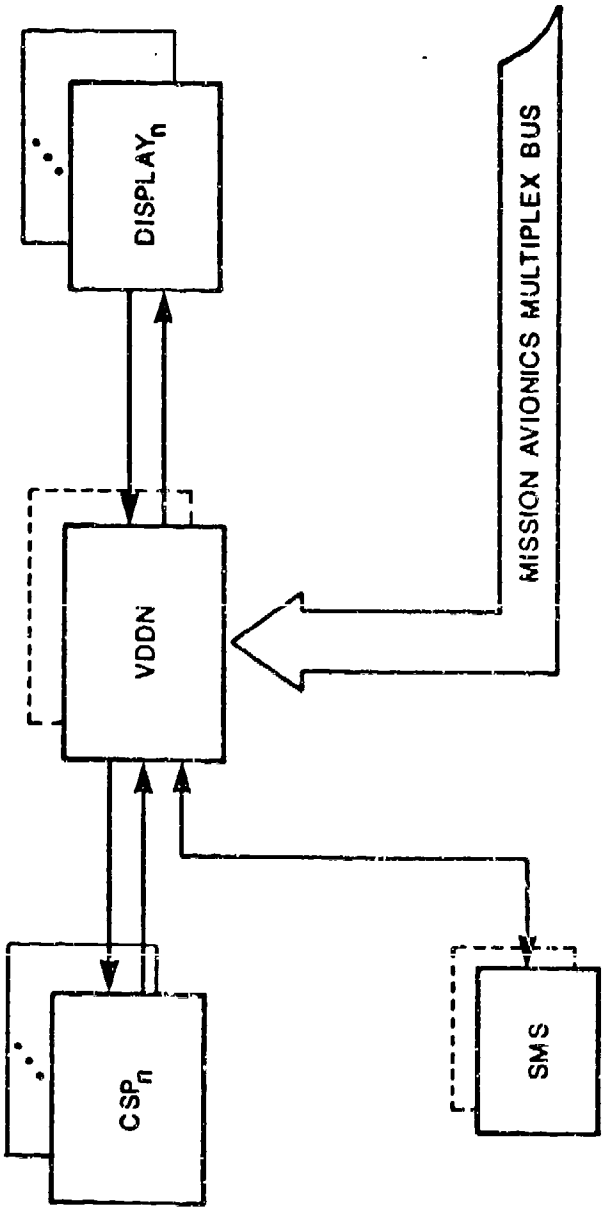


Figure 3.7.10-1 VIDEO DATA DISTRIBUTION NETWORK

failure or battle damage, the System Executive will provide the pilot with a report of any planned functional capabilities that will be unavailable as a result of the failure. The System Executive functions/interfaces are depicted in Figure 3.7.11-1.

3.7.12 Kernel Executive

The Kernal Executive is responsible for controlling the application software assigned to a processor, providing control and communication between application tasks, controlling peripheral devices, and participating in processor level fault tolerance operations. Control of the application software is provided by a task sequence which handles the invocation and suspension of tasks within the processor. Interim task control and communication services are provided to support the Ada (MIL-STD-1815A) tasking model which includes the concept of a rendezvous between tasks, allowing intertask control and communication. The peripheral device control capability of the Kernel Executive permits an application task to request an operation on a peripheral device at a logical level. This function is provided to allow the application software executing in the data processor portion of the signal processors the capability of controlling the execution of the common signal processing modules. Processor level fault tolerance operations include the receipt of processor error indicators, analysis of the conditions, and determination of the appropriate action. This appropriate action may be to pass the condition to an application task for further action by an exception handler, notify the system executive, file the processor, or any continuation of these or other actions.

All operating system services required by the application software are handled through the Kernel Executive which in turn will access services provided by the Distributed Executive as required. The Kernel Executive functions/interfaces are depicted in Figure 3.7.12-1.

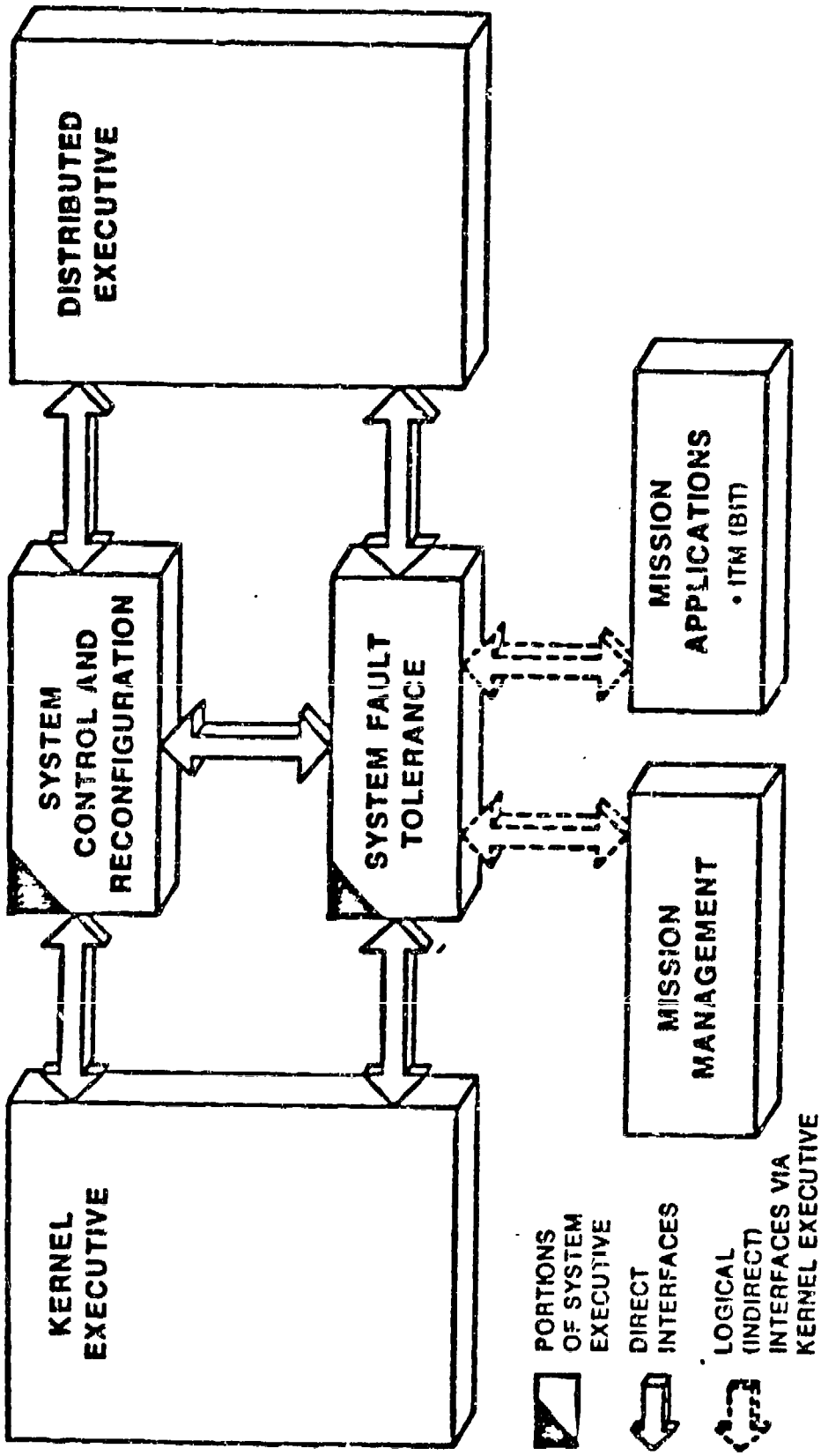


Figure 3.7.11-1 SYSTEM EXECUTIVE

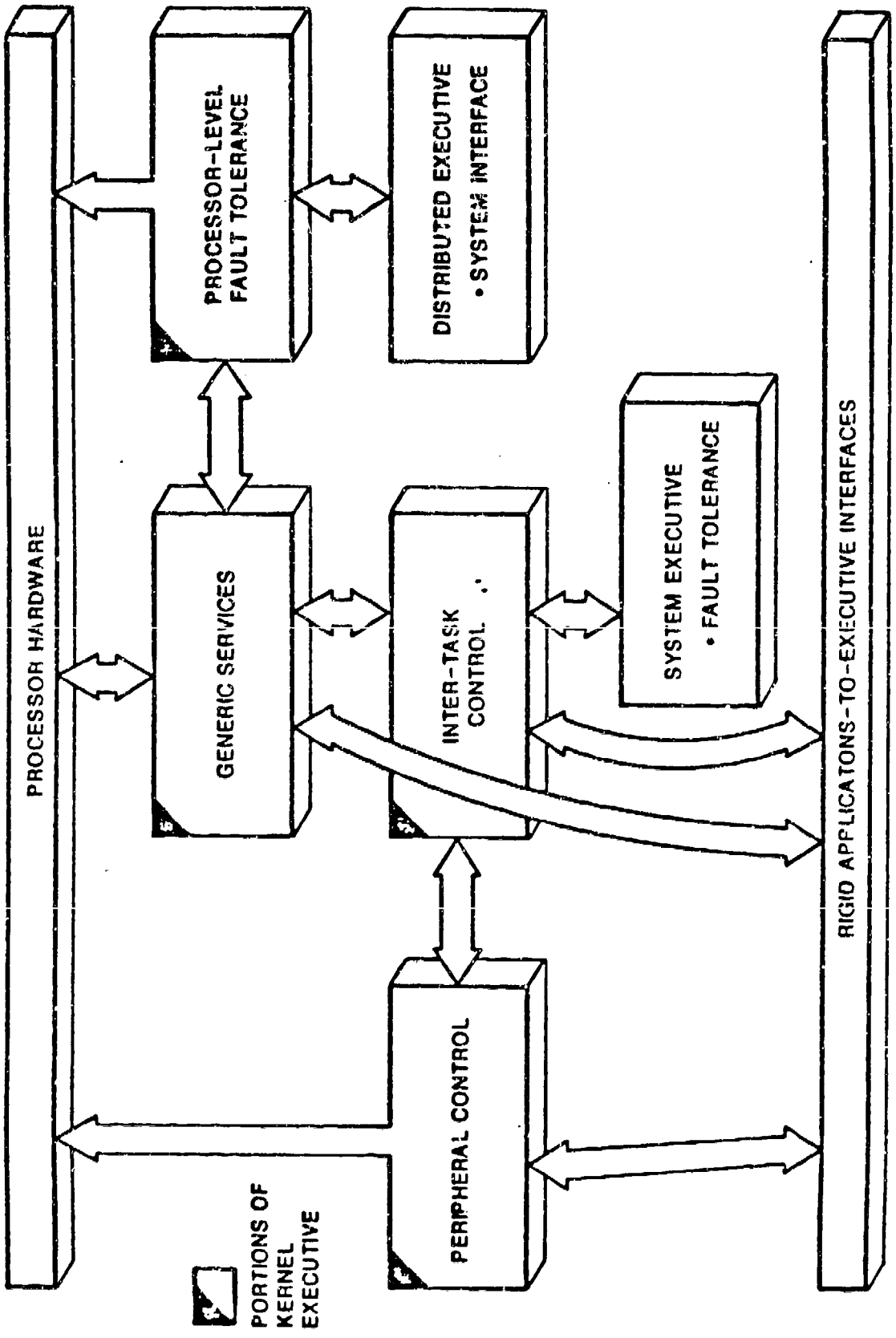


Figure 3.7.12-1 KERNEL EXECUTIVE

3.7.13 Distributed Executive

The Distributed Executive provides bus control interface capability, inter-processor data transfer operations, and general participation in the overall system operation. The Distributed Executive communication interface provides for the exchange of information between processors. Data transfer requirements will result from transactional requirements in support of application tasks and from system status monitoring and reconfiguration operations. System software and hardware configuration information shall be maintained by every processor in the system. The Distributed Executive is responsible for the maintenance of the system state information requirements and system operation. The Distributed Executive interfaces with the Distributed Executive in other processors via the Mission Avionics Multiplex bus and with the Kernel Executive resident within its own processor. The Distributed Executive functions/interfaces are depicted in Figure 3.7.13-1.

4.0 QUALITY ASSURANCE PROVISIONS (Not Applicable)

5.0 PREPARATION FOR DELIVERY (Not Applicable)

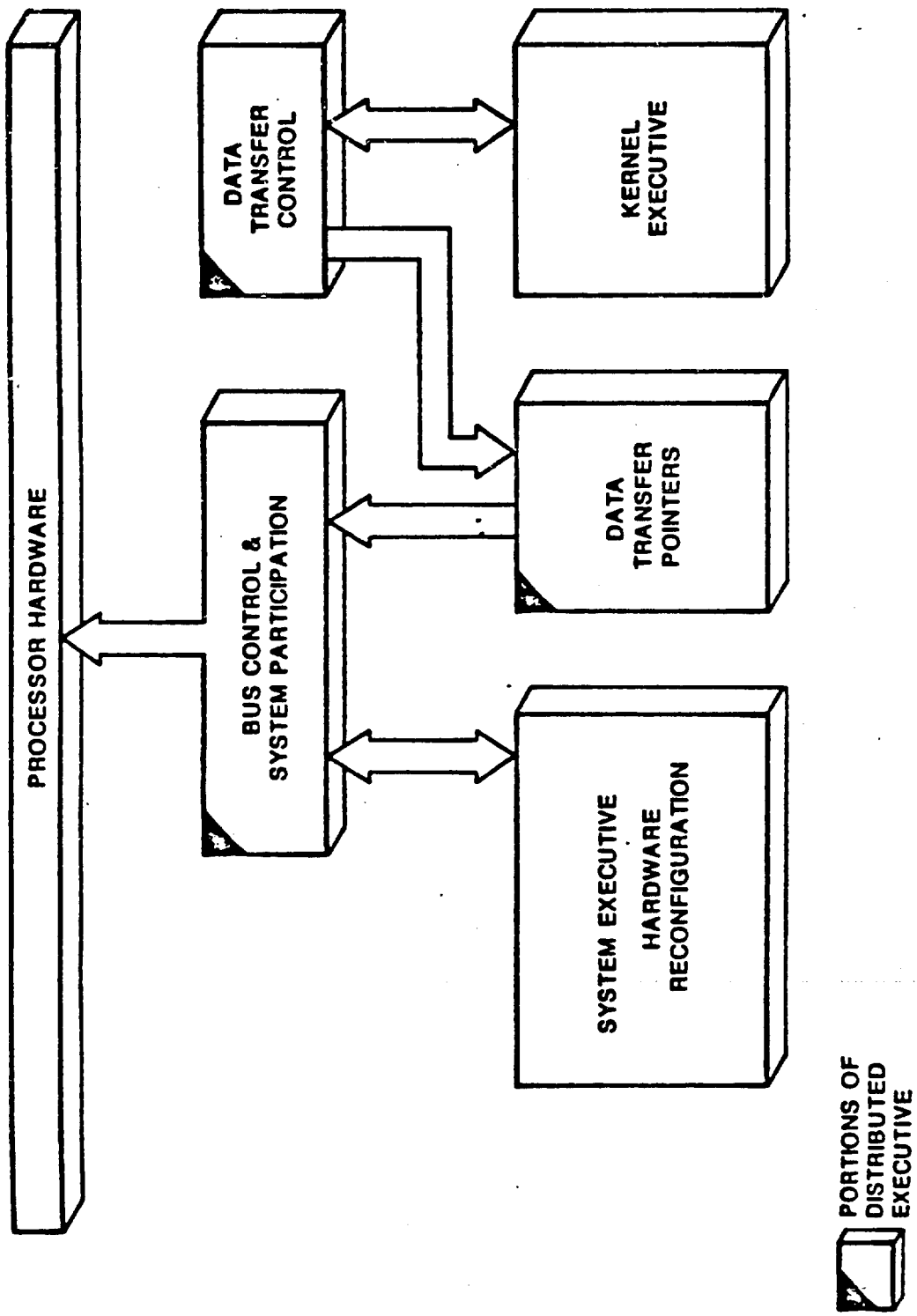


Figure 3.7.13-1 DISTRIBUTED EXECUTIVE

6.0 NOTES

6.1 Common Module Set

The maximum use of common line replacable modules across the entire avionics suite is a basic design requirement for the Pave Pillar system. Common modules are used to perform common functions throughout the system. The following subparagraphs describe a representative set of Common Modules for use in the Pave Pillar architecture and for use in associated subsystems. They are based on the VHSIC 1750A and Common Signal Processor ADM module sets under development as well as perceived needs for additional modules not currently under development under those programs. Following the core ADM set of modules, (1-15) additional modules are specified which are not constrained by the implementation approaches currently being used on the V1750A and CSP projects.

In the signal processing area, a Module Functional Group is identified which consists of one controller module (Element Supervisor Unit) and one, two, or more functional element modules (Global Memories, Processing Elements, or I/O modules) that are locally controlled by that Element Supervisor Unit (ESU). This Module Group can also be viewed as a type of Supermodule with three standard interfaces: the PI-Bus, the TM-Bus, and the Data Flow Network interface. With VHSIC Phase 1 circuitry it usually is not possible to contain a complete signal processing module group on a single module; hence several modules are necessary for CSP functions. This necessitates local interfaces between modules which need to be standardized for a given family of modules. Within CSP two local buses are specified, the Element Control Bus (ECB) and the Element Maintenance Bus (EMB) which currently are unique to the CSP set of modules. In the future, it is expected that these local buses will be absorbed into the module.

Three power modules are included which provide the Pave Pillar architecture with a set of modules capable of powering a variety of system configurations. The power modules conform to the SEM E form factor. The DC-DC convertor modules are distributed throughout the system backplane, which improves the power system's transient response. The modules are connected in parallel and share the load equally, allowing the addition of the appropriate number of spares to the system. Built-In-Test features report module faults to the system, warning the user to avoid critical modes and reducing the maintenance support. The power system shall include Nuclear Event Detection (NED) and Electromagnetic Pulse (EMP) circumvention. The power system shall have an overall power density of 4 watts/cubic inch, an efficiency of 70% to 80%, and a weight efficiency of 0.02 pounds/watt. The following sub-paragraphs describe a set of common modules for use in the Pave Pillar Architecture. Table 6.1-1 lists the module types which form this baseline set.

6.1.1 VHSIC 1750A Processor Module (V1750A)

The 1750A Processor Module will consist of a 1750A Instruction Set Architecture (ISA) CPU with 256K words of dedicated local memory and shall contain two PI-Bus interfaces, a Test and Maintenance Bus interface and an IEEE 488 Bus interface. The 1750A Processor Module shall be capable of a minimum of 3 Million Instructions Per Second (MIPS) assuming a DAIS MIL-STD-1750A weighted instruction mix without extended memory addressing capability enabled and a minimum of 2 MIPS with extended memory addressing enabled. The 1750A Processor Module shall contain internally (in non-volatile memory) a 4K word start-up ROM for storing initialization and/or Kernel Executive software. The Non-Volatile 1750A Processor Module shall be functionally identical, except that its program storage memory (ROM) shall not be alterable in flight. In addition, since the module shall not perform self loading, the

TABLE 6.1-1
BASELINE COMMON MODULE SET

1. VHSIC 1750A CPU Module (V1750A)
2. Avionics Bus Interface Module (ABI) Module
3. Bulk Memory Module (BMM)/Non-volatile Bulk Memory Module (NVBMM)
4. MIL-STD-1553B I/O Module (V1553B)
5. Dedicated Interface Modules
   - A/D Conversion
   - D/A Conversion
   - Discretes In
   - Discretes Out
   - Serial Channel (In & Out)
6. Floating Point Processing Element (FPPE) Module
7. Global Memory (GM) Module
8. Data Network Element (DNE) Module
9. Sensor Interface (SI) Module
10. Element Supervisor Unit (ESU) Module
11. Timing and Control Generator (TCG) Module
12. I/O Terminator Module
13. DC-DC +5 Volt Power Converter Module
14. DC-DC Multi-voltage Power Converter Module
15. Power Conditioner Module
16. Fixed Point Vector Processing Element (VPE) Module
17. Electronic Warfare-oriented Processing Element Modules
18. Biphase Correlator Processing Element (BCPE) Module
19. MIL-STD-1760 Stores I/O Module
20. Video Module
21. Display Module
22. Network Switch Module (NSM)
23. Key Generator Module (KGM)

internal Kernel Executive software shall be unique. The 1750A Processor Module is shown in Figure 6.1.1-1.

6.1.2 Avionics Bus Interface (ABI) Module

The Avionics Bus Interface (ABI) Module shall be designed to interface with either the Mission Avionics Multiplex Bus, the Block Transfer Multiplex Bus, or the Vehicle Management System Multiplex Bus. The ABI shall contain two PI-Bus interfaces and a test and Maintenance Bus interface. The ABI shall provide for redundant media and shall provide redundant data paths within the module. The ABI Module is shown in Figure 6.1.2-1.

6.1.3 Bulk Memory Module (BMM)/Non-Volatile Bulk Memory Module (NVBMM)

The Bulk Memory Module shall be capable of storing 2 million words of Random Access Memory. There shall be two versions of this Module. One version shall contain volatile memory and the other version shall contain Non-Volatile Memory. The Module shall contain two PI-Bus interfaces and a Test and Maintenance Bus interface. The Bulk Memory Module shall utilize a single error correction/double error detection scheme. The Bulk Memory Module is shown in Figure 6.1.3-1.

6.1.4 MIL-STD-1553B I/O Module

The MIL-STD-1553B I/O Module shall provide a dual redundant MIL-STD-1553B Multiplex Bus interface terminal. The bus terminal shall be programmable to operate in either Master or Remote mode. The module will have two PI-Bus interfaces and a test and Maintenance Bus interface. The MIL-STD-1553B I/O Module is shown in Figure 6.1.4-1.

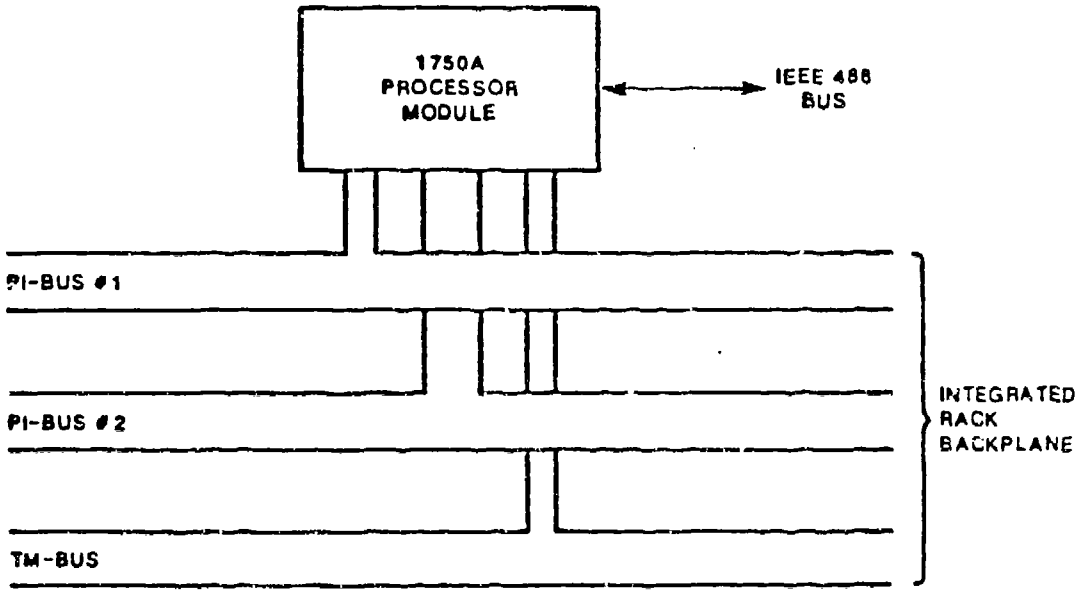


Figure 6.1.1-1 1750A PROCESSOR MODULE

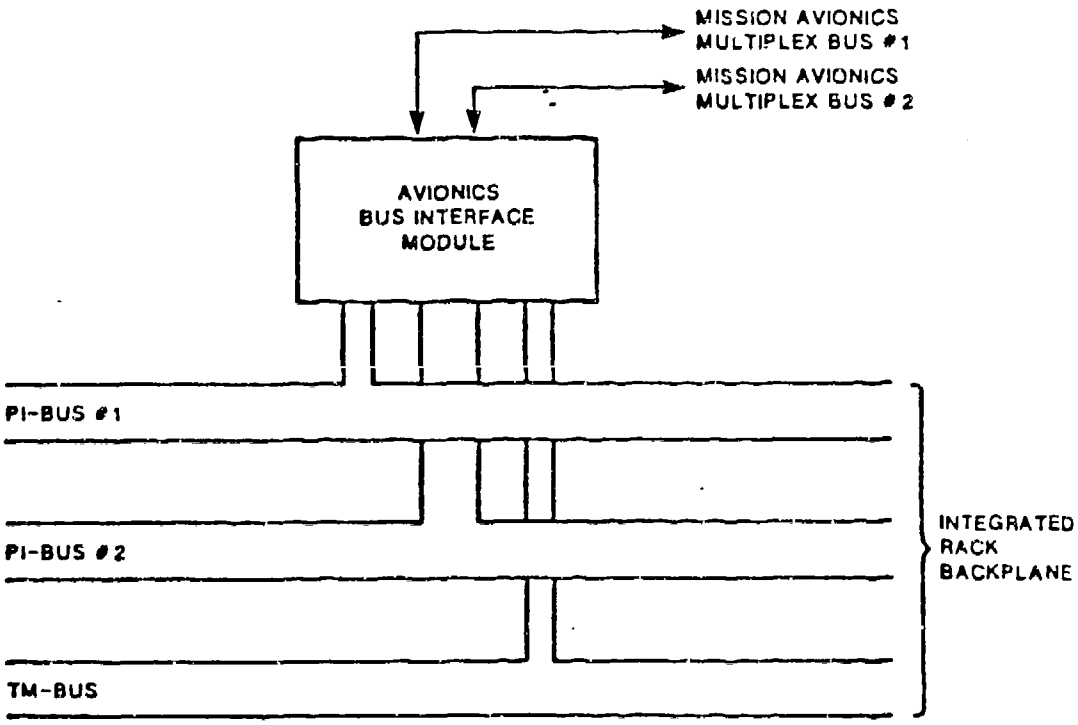


Figure 6.1.2-1 AVIONICS BUS INTERFACE MODULE

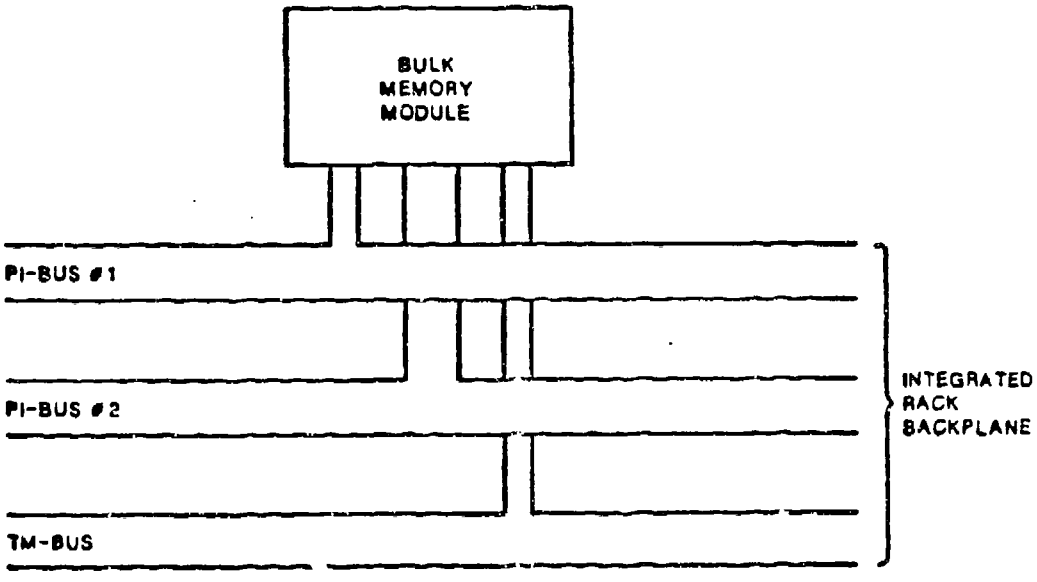


Figure 6.1.3-1 BULK MEMORY MODULE

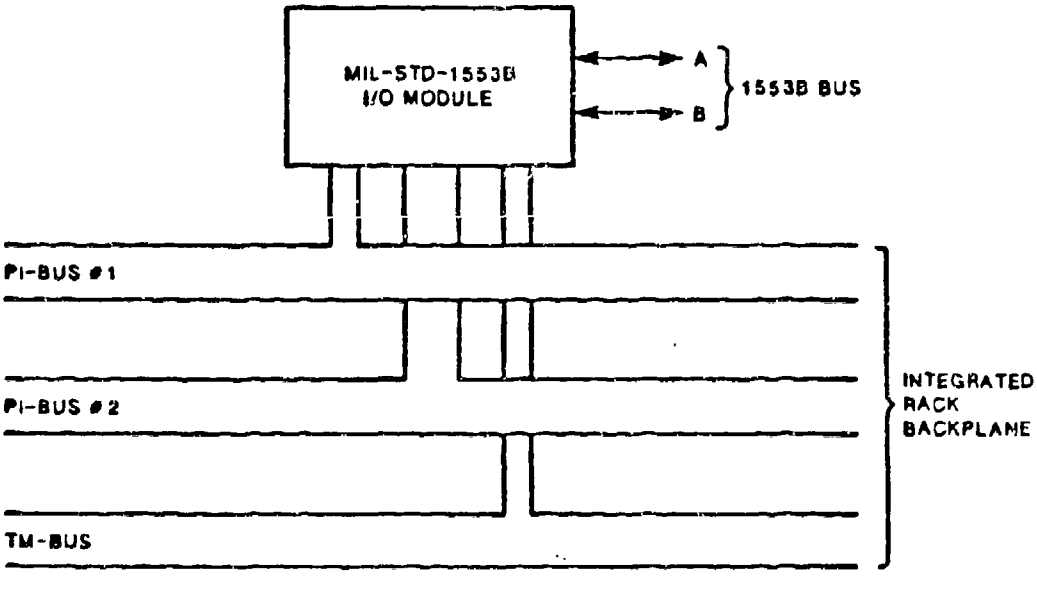


Figure 6.1.4-1 MIL-STD-1553B I/O MODULE

6.1.5 Dedicated Interface Module

The Dedicated Interface Module shall provide the buffering and control circuitry to manage discrete interface signals, both input and output, such as analog to digital, digital to analog, synchro, serial channels, TTL level, etc. The Dedicated Interface Module may be designed as several mixes of signal types per module. The Dedicated Interface Module will have two PI-Bus interfaces and a test and Maintenance Bus interface. The Dedicated Interface Module is shown in Figure 6.1.5-1.

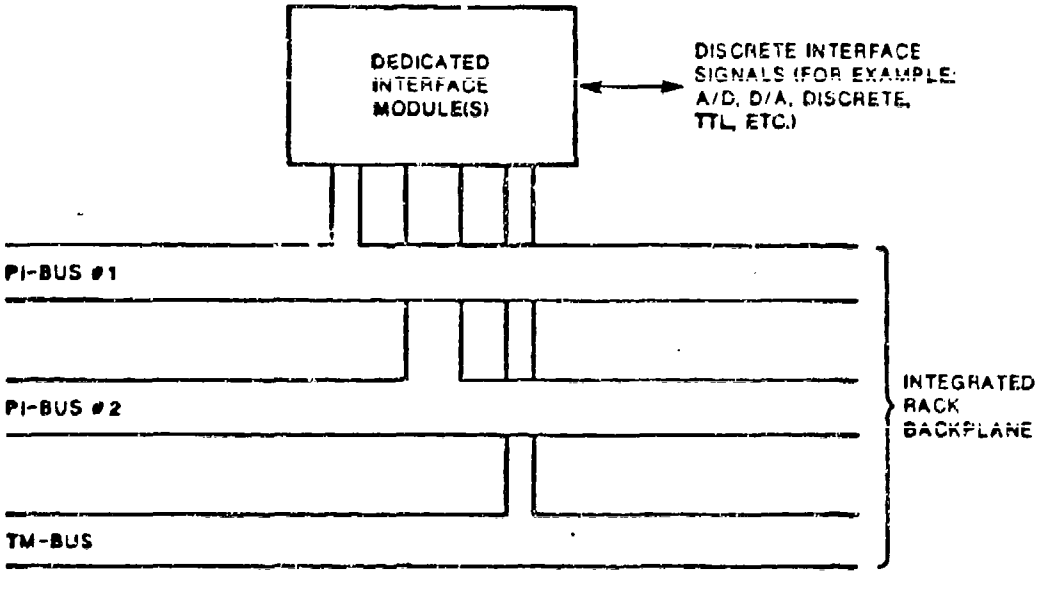


Figure 6.1.5-1 DEDICATED INTERFACE MODULE

6.1.6 Floating Point Processing Element (FPPE) Module

The Floating Point Processing Element (FPPE) shall provide the capabilities to process both 32-bit floating point and 16- and 32-bit fixed point arithmetic efficiently for signal processing computations. This module shall be able to perform 125 million floating point operations per second or 12.5 million complex floating point operations per second. The module shall support floating and fixed point multiplication, addition, subtraction, comparisons, data dependent operations and data conversions. It shall also support logical operations, table look-up functions, square root and reciprocal calculations, sine/cosine generation, arctangent generation, optionally selected data clamping, optionally selected status reporting, and error reporting. The FPPE shall permit overlapped task execution with task loading and unloading. It shall include local memories for data and coefficient storage of at least 16K 32-bit data words visible during task execution. The FPPE shall include the capability to check for a variety of internal errors, including parity errors, addressing errors, and calculation errors. Detected errors shall be logged in a non-volatile fault log memory device for later access over the Element Maintenance Bus (EMB).

The FPPE shall have 3 external interfaces:

a. Element Control Bus (ECB), ICD 2875826

b. Element Maintenance Bus (EMB), ICD 2875785

c. Data Flow Network (DFN) Interface, ICD 2875788

The ECB is a local control bus from the Element Supervisor Unit (ESU) which controls the FPPE; it selects which signal processing Macros to execute, establishes data network paths and messages, etc. The EMB is a local maintenance bus for

testing and fault isolating the FPPE from the ESU. The DFN port provides high speed access to sensor data stored elsewhere in the system, such as in Global Memories, or arriving over Sensor Interfaces (SIs). The FPPE Module is shown in Figure 6.1.6-1.

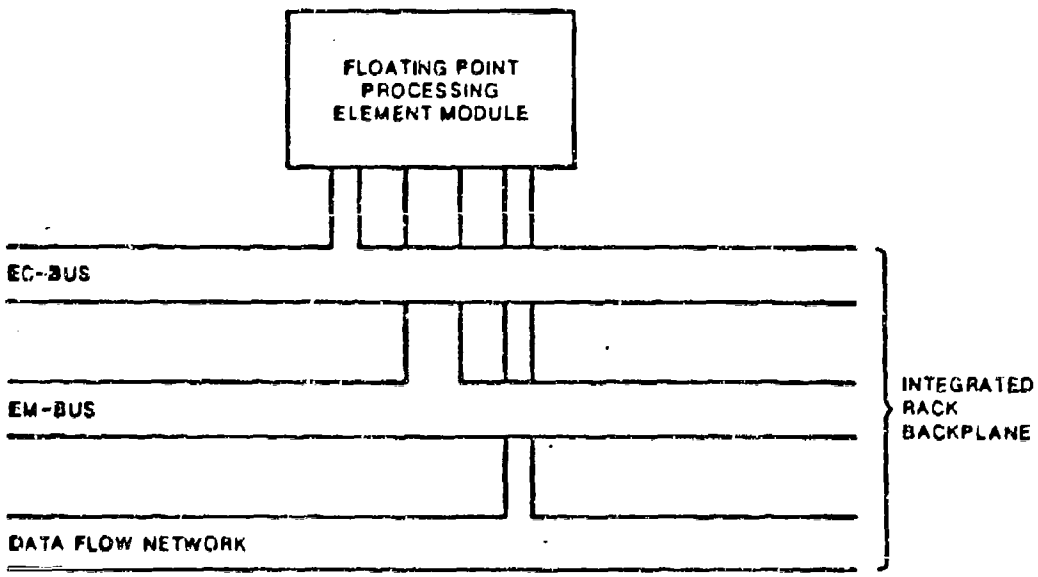


Figure 6.1.6-1 FLOATING POINT PROCESSING ELEMENT MODULE

6.1.7 Global Memory (GM) Module

The Global Memory Module shall be capable of storing 1 million 32-bit data words of volatile memory. The module shall support complex address modes needed for signal processing, including the following:

a. Circular Queue and Multiple Circular Queues

b. Corner Turn Queues

c. Coordinate Rotation Queues

d. Random Access Queue

e. Buffer

These address modes shall be supported at a burst transfer rate of 25 million 32-bit words per second. The memory shall support logical to physical address translation in order to prevent memory fragmentation. The mapping scheme shall use 4K block sizes and permit a logical space larger than physical memory to aid in memory management.

The GM shall include error detection and correction circuitry on the main memory. The implementation shall be 8 EDC bits that correct in 4-bit groups (4-bit wide memory circuits are specified, hence error patterns of 4 bits are likely). The GM shall include a non-volatile fault log memory device for recording detected faults for later maintenance.

The GM shall have 3 external interfaces:

a. Element Control Bus (ECB), ICD 2875826

b. Element Maintenance Bus (EMB), ICD 2375785

c. Data Flow Network (DFN) Interface, ICD 2875788

The ECB is a local control bus from the Element Supervisor Unit (ESU) which permits the ESU to control the GM and provides a path for the GM to access ESU main memory and the PI-Bus interface. The ECB shall permit transfers from ESU elements to Global Memory, GM address translation memory, and the Data Flow Network. The EMB is a local maintenance bus for testing and fault isolating the GM from the ESU. The DFN port provides high speed access to data stored in the GM to other elements in the signal processing function, such as FPPEs and SIs. The Global Memory Module is shown in Figure 6.1.7-1.

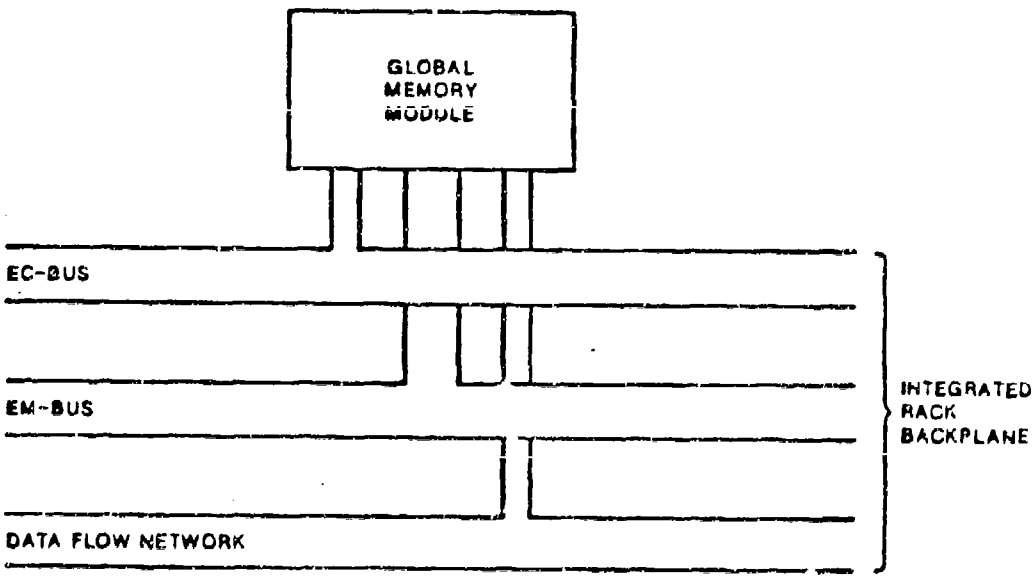


Figure 6.1.7-1 GLOBAL MEMORY MODULE

6.1.8 Data Network Element (DNE) Module

The DNE Module shall permit the implementation of a variety of configurations of Data Flow Networks with respect to the number of ports and the width of the data words transferred. The baseline Data Flow Network for the signal processing function has 24 ports using 32-bit data word transfers; it permits 12 simultaneous paths to be operating between the 24 ports. This configuration requires 8 DNE Modules. The DNE Module shall have 12 ports each with 16 data lines. The DNE shall be able to transfer between up to 6 sets of external Master/Slave pairs of elements connected to the module simultaneously. The ports shall operate in half duplex mode: data can transfer in either direction, but only one direction at a time. Once a path is established, transfers shall take place at a 25 million word rate. The operation of the path shall conform to the interface design specification for the Data Network, ICD 2875788. The DNE shall provide error checking for transfers over the Data Flow Network. Odd parity shall be provided on each 16-bit data segment. A non-volatile fault log memory circuit shall be included on the DNE to log any DNE faults.

The DNE Module shall have two external interfaces:

a. The Data Flow Network ports as documented by ICD 2876788.

b. The Element Maintenance Bus (EMB) as documented by ICD 2875785.

The EMB shall permit an ESU to access the fault log memory device and determine the health of the DNE Module. The DNE Module is shown in Figure 6.1.8-1.

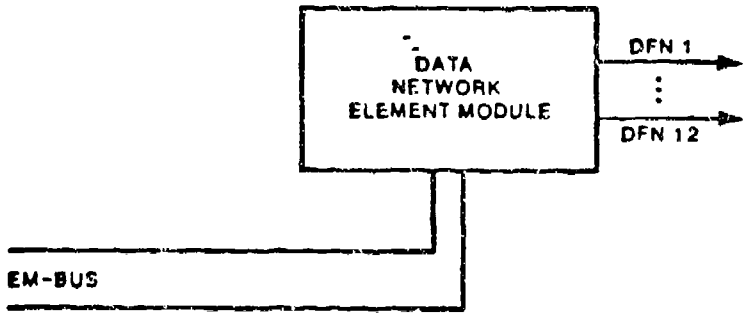


Figure 6.1.8-1 DATA NETWORK ELEMENT MODULE

6.1.9 Sensor Interface (SI) Module

The SI Module provides a gateway between internal signal processing group data paths and an external sensor data distribution network. The principal gateway is between the Data Flow Network and the external sensor channel; paths also exist between the Element Supervisor Unit (ESU) local memory and the sensor channel and the ESU local memory and the Data Flow Network. The ESU connects to the PI-Bus and TM-Bus, thus permitting transfer between those buses and the other paths. The external interfaces to the SI Module are:

a. Element Control Bus (ECB) documented by ICD 2875826

b. Element Maintenance Bus (EMB) documented by ICD 2875785

c. Data Flow Network (DFN) documented by ICD 2875788

d. Sensor Data Distribution Network (SDDN).

The ECB permits the ESU controller module to manage the SI module. The ESU initializes the SI, fields interrupts and reads status, and permits SI access to message control blocks via DMA transfers from the ESU local memory. The ECB path permits PI-Bus to SI transfers which in turn may be forwarded over the Data Flow Network or the external Sensor Data Distribution Network.

The EMB permits the ESU to initiate SI self-test and read the resulting status. It permits access to a non-volatile fault log memory on the SI storing any detected faults.

The Data Flow Network port on the SI is the principal data transfer path to the SI from the signal processor modules. It is used for sensor data transfers from sensor front-end units to Global Memories or processing elements. It can also transfer data from signal processing modules to front-end units.

The SI port to the external sensor data distribution network permits sensor data to be input to the signal processing modules. It shall have a burst transfer rate of 500 million bits per second. It shall support a point-to-point transfer configuration. The baseline physical link layer shall be implemented with fiber optics (alternate media may be appropriate for near-term application such as wire provided the performance requirements are met). The design shall also support other configurations such as a ring bus. Error detection shall be provided on the sensor data distribution channel.

The SI module shall support overlapped transfers from the front-end units and the signal processing Data Flow Network so that continuous data streams may be accommodated. Local buffering on the module shall consist of at least three 4K x 34 data buffers to support overlapped operations. The SI Module is shown in Figure 6.1.9-1.

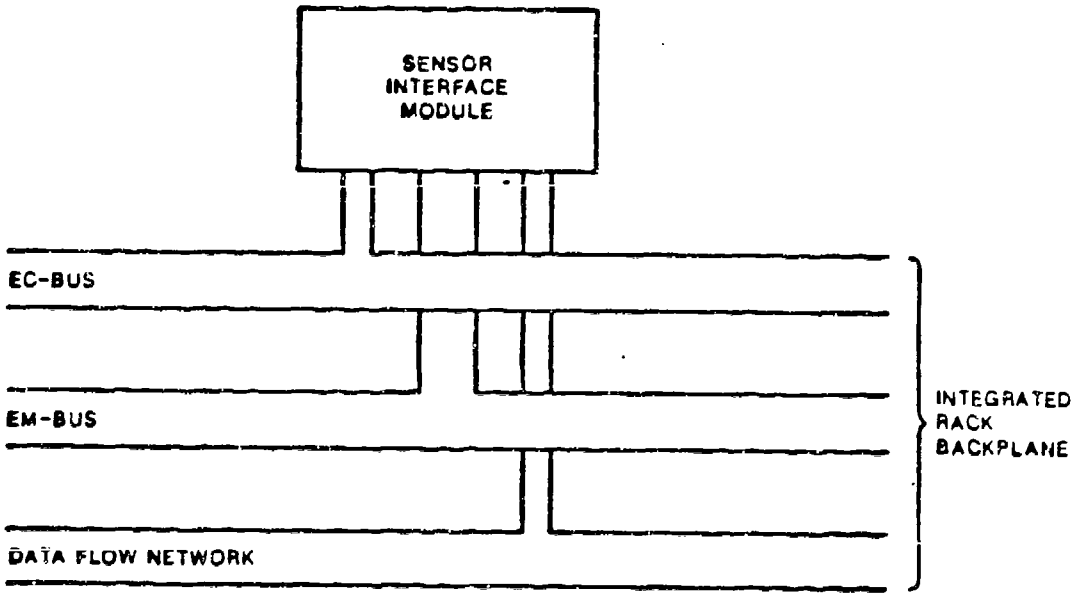


Figure 6.1.9-1 SENSOR INTERFACE MODULE

6.1.10 Element Supervisor Unit (ESU) Module

The ESU Module is the controller module for signal processor processing element (PE) modules, Global Memory (GM) modules, and interface modules. It shall perform the control function for the Floating Point Processing Element (FPPE), the fixed point Vector Processing Element (VPE), the Biphase Correlator Processing Element (BCPE), the low latency General Purpose Processing Element (GPPE), the GMs, the Sensor Intefaces (SIs), and the COMSEC interface (CI). It performs the maintenance function for the Data Network Element (DNE).

The ESU shall have the following external interfaces:

a. PI-Bus.
b. TM-Bus.
c. Element Control Bus (ECB).
d. Element Maintenance Bus (EMB).

The PI-Bus (See Appendix B - VHSIC Phase 2 Interoperability Standard - PI-Bus), provides a path for the signal processing module "group" controlled by the ESU to communicate with the rest of the signal and data processing system. It is used primarily for control messages. The PI-Bus interface on the ESU shall provide autonomous DMA capability to local ESU memory.

The TM-Bus (See Appendix C - VHSIC Phase 2 Interoperability Standard - TM-Bus), provides a path for test and maintenance commands and data to be sent to the ESU and the modules it controls.

The ESU shall implement a hierarchical maintenance bus system with the Element Maintenance Bus (EMB), documented in ICD 2875785, connecting the ESU to the modules it controls.

The Element Control Bus (ECB) is the local bus by which the ESU controls the modules in its group. It is used for initializing modules, sending commands and data, requesting status, fielding interrupts, and permitting the attached modules DMA access to ESU local memory for Control Blocks defining local operations such as signal processing Macros or Data Flow Network message transfers.

The ESU shall implement the MIL-STD-1750A Instruction Set Architecture (ISA). It shall have a throughput of at least 2 million instructions per second against the DAIS instruction mix. The local memory shall be at least 256K 16-bit words; memory management shall be implemented. Timers A and B shall be implemented. A Start-Up ROM shall be included for module initialization.

The ESU shall include fault detection and isolation capabilities. Parity checking shall be provided on local memory and the ECB; a non-volatile fault log memory shall be included accessible from the TM-Bus. The ESU Module is shown in Figure 6.1.10-1.

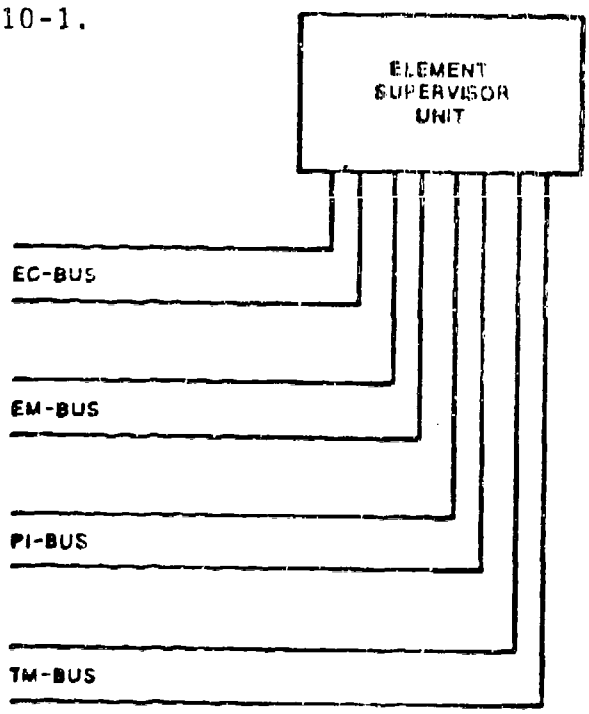


Figure 6.1.10-1 ELEMENT SUPERVISOR UNIT MODULE

6.1.11 Timing and Control Generator (TCG) Module

The TCG module shall provide the system clocks needed by signal and data processing core modules. It shall also contain a system fault log non-volatile memory for recording faults for modules or components not having fault log memories. It accepts a Power On Reset signal from the power supply system, software, or an external source and generates a system reset signal. The clocks provided by the TCG shall include:

a. 25 MHz for basic VHSIC Phase 1 circuitry use.

b. 12.5 MHz for PI-Bus clocking.

c. 6.25 MHz for TM-Bus clocking.

Because of skewing problems, individual signals for each module shall be generated for distribution. The TCG Module is shown in Figure 6.1.11-1.

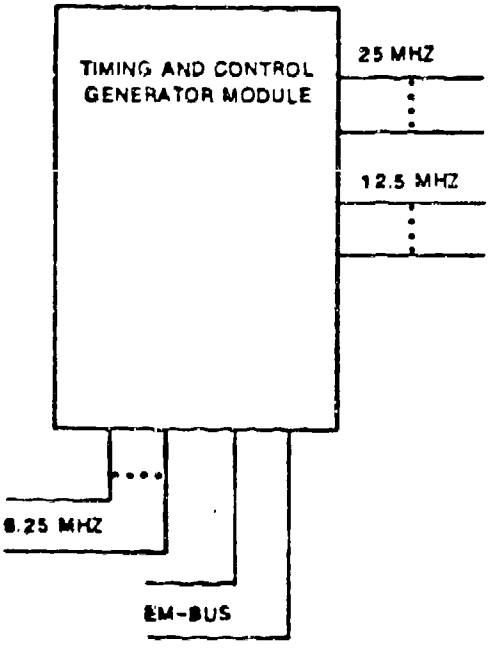


Figure 6.1.11-1 TIMING AND CONTROL GENERATOR (TCG) MODULE

6.1.12 I/O Terminator Module

The I/O Terminator Module shall provide I/O termination for the PI-Bus, the TM-Bus, and any other system bus needing termination. The I/O Terminator Module is shown in Figure 6.1.12-1.

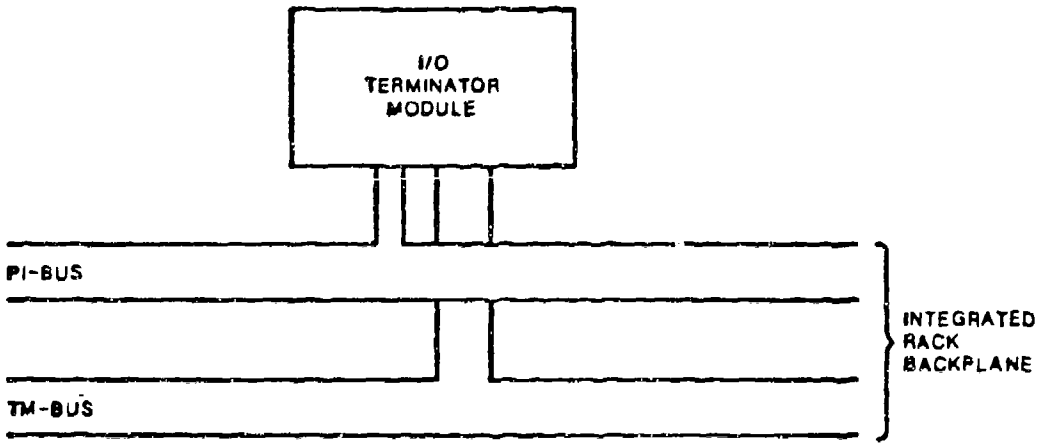


Figure 6.1.12-1 I/O TERMINATOR MODULE

6.1.13 DC-DC 5 Volt Power Converter Module

The 5 volt power converter module shall be capable of supplying 200 watts with +/-5% regulation. It shall use +/-24 VDC as input. It shall occupy a single 0.6 inch module slot on the backplane. It shall include sense circuits for detecting if the module is supplying power, if the voltage is within regulation, and if the module's temperature is within its thermal limit. The sense outputs shall be available at a BIT interface, baselined as discretes signals. (The Power Conditioner module is specified to be a central collection point for BIT discretes for the distributed converter modules.) The module 5 volt output is through the module connector to the backplane which distributes the voltage to user modules.

Fuse links shall disconnect a module from the system in the event a module's output shorts. Overvoltage protection will prevent damage to other modules. The DC-DC 5 Volt Power Convertor Module is shown in Figure 6.1.13-1.

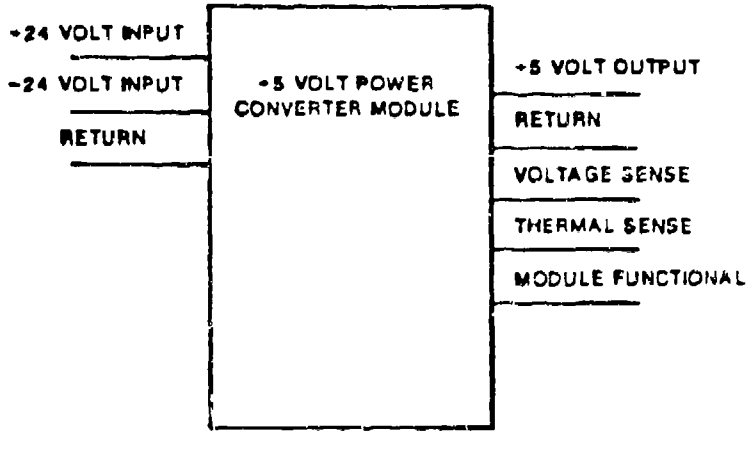


Figure 6.1.13-1 **DC-DC +5 V POWER CONVERTER MODULE**

6.1.14 DC-DC Multi-Voltage Power Converter Module

The multi-voltage converter module shall provide +/-15, +3.3, and +2 volts. It shall require a +/-24 volt input. It shall occupy a single 0.6 inch slot. It shall include sense circuits for detecting if the module is supplying power, if the voltage is within regulation, and if the module's temperature is within its thermal limit. The sense outputs shall be available at a BIT interface, baselined as discretes signals. (The Power Conditioner module is specified to be a central collection point for BIT discretes for the distributed converter modules.) The module outputs are brought out through the module connector to the backplane which distributes them to user modules. Potential future requirements of +/-12, -2.0, and -5.2 volts have been identified for the High Speed Data Bus (HSDB) interface. The multi-voltage module shall be adaptable to be able to provide these voltages with modest current capabilities. Fuse links shall disconnect a module from the system in the event a module's output shorts. Overvoltage protection will prevent damage to other modules. The DC-DC Multi-Voltage Power Converter Module is shown in Figure 6.1.14-1.

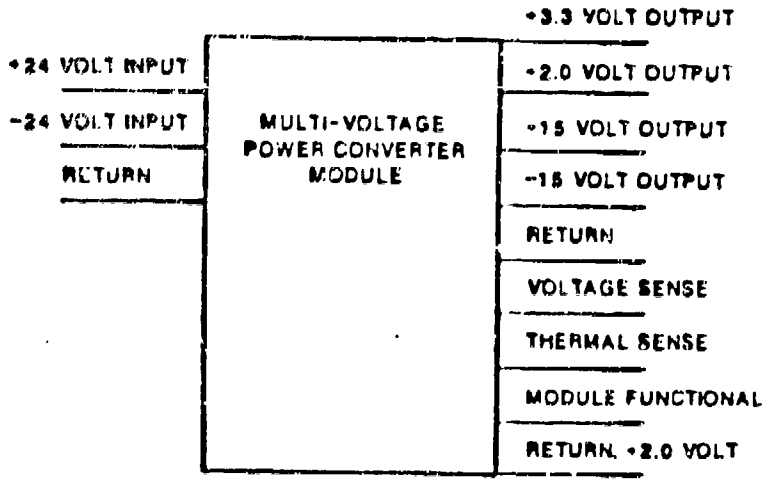


Figure 6.1.14-1 **DC-DC MULTI-VOLTAGE POWER CONVERTER MODULE**

6.1.15 Power Conditioner Module

The Power Conditioner Module filters the aircraft power bus and converts it to +/-24 VDC for distribution to the DC-DC converter modules. It shall be able to accept input power from either a 115 VAC 400 Hz 3-phase 4 wire Wye power source or a 270 VDC power source. Sense circuits shall be included to detect if the module is outputting power, if the output is within +/-10% regulation, and if the module's temperature is within its thermal limit. These BIT discrete signals shall be available to the system over a TM-Bus interface. In addition, the Power Conditioner shall log any detected faults in a non-volatile memory device within the Power Conditioner for later maintenance use. The baseline design includes the ability to accept BIT discrete inputs from the distributed DC-DC converters for output over the TM-Bus to the system and logging within the Power Conditioner. Fuse links will disconnect the module from the system in the event its output shorts. Overvoltage protection shall be provided to prevent damage to other modules. The Power Conditioner shall maintain enough stored energy to provide power through a 50 microsecond prime power loss. Distribution of the +/-24 VDC is through separate parallel conductors connecting to the distributed converter modules independent of the backplane. The Power Module is shown in Figure 6.1.15-1.

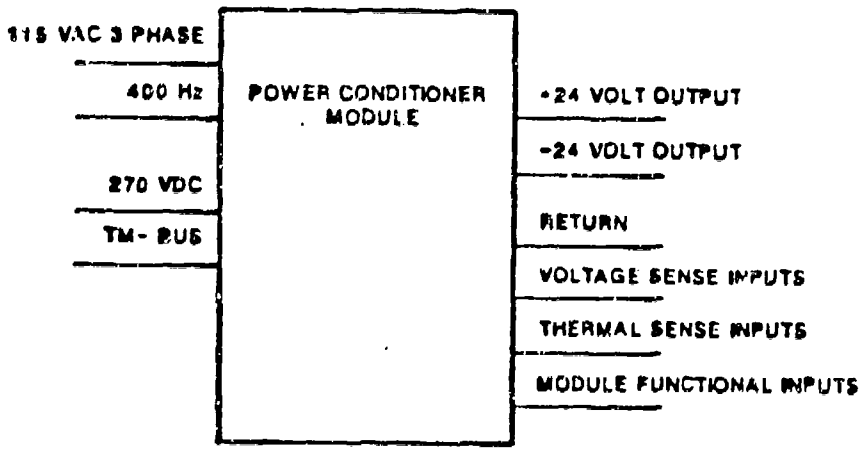


Figure 6.1.15-1 **POWER CONDITIONER MODULE**

6.1.16 Fixed Point Vector Processing Element (VPE) Module

The VPE module is a 16-bit fixed point arithmetic vector-oriented processing element that operates efficiently on complex (I and Q) data arranged in vectors or matrices. Its arithmetic support is augmented for complex multiplys, adds, and subtracts to provide for very fast FFT calculations, convolutions, correlations, etc. It is intended to perform computationally intensive radar signal processing functions such as pulse compression and adaptive sidelobe cancellation perhaps three times as fast as a FPPE. (One VPE could replace 3 FPPEs in a radar application for Track-While-Scan modes, for instance.) The VPE shall include fault detection and isolation logic, including parity check circuits, arithmetic error checking, etc, and include a non-volatile fault log memory device. The VPE shall meet the standard signal processing PE interfaces:

a. Element Control Bus (ECB)
b. Element Maintenance Bus (EMB)
c. Data Flow Network (DFN).

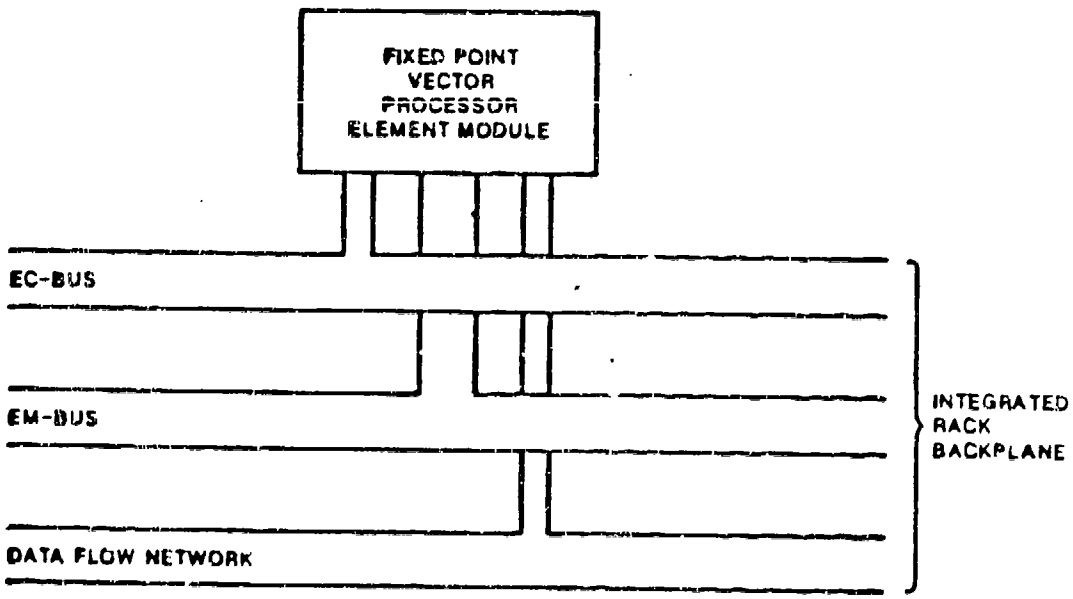


Figure 6.1.16-1 **FIXED POINT VECTOR PROCESSING ELEMENT MODULE**

6.1.17 Electronic Warfare-oriented Processing Element Modules

The Electronic Warfare functional area requires digital sorting and filtering functions with very high throughput requirements at the "front-end" of the EW processing load. The detailed requirements of this Sort Enhanced Processing Element (SEPE) module are TBD. A low latency, General-Purpose Processing Element (GPPE) is also needed in the EW area. This module shall implement a 32-bit Reduced Instruction Set Architecture (RISA) with a target throughput goal of 10 million instructions per second. It shall have a very fast I/O instruction capability to meet short response I/O requirements; a 2-clock I/O instruction capability is a goal. It shall directly address large memory sizes. It shall interface to the Data Flow Network, the PI-Bus, and the TM-Bus. It shall include a Fault Log memory device for maintenance purposes. The EW Processing Element Module is shown in Figure 6.1.17-1.

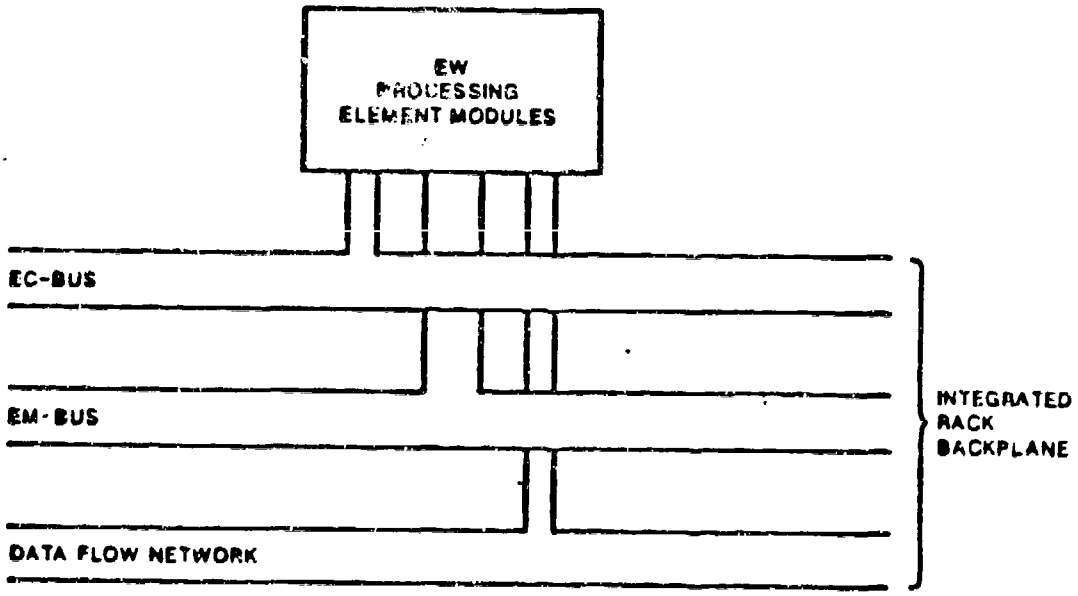


Figure 6.1.17-1 **EW PROCESSING ELEMENT MODULES**

6.1.18 Biphase Correlator Processing Element (BCPE) Module

The Biphase Correlator Processing Element (BCPE) shall be designed to efficiently compute biphase correlations of signals with 2 to 16 bits of accuracy and 16 to 512 points. The BCPE throughput shall be adequate to perform the CNI broadband acquisition correlation functions for JTIDS, GPS, etc, and the radar (URR) pulse compression functions when biphase methods are used. The BCPE shall contain a minimum of two 8K banks of local memory that permit overlapped I/O and computation. The BCPE shall have 3 external interfaces:

a. Element Control Bus (ECB)

b. Element Maintenance Bus (EMB)

c. Data Flow Network (DFN)

(See paragraph 6.1.6 FPPE Module for a description of these three interfaces.)

The BCPE shall include fault detection and isolation capabilities for internal memories, buses, and arithmetic elements. It shall include a non-volatile fault log memory. The BCPE Module is shown in Figure 6.1.18-1.

6.1.19 MIL-STD-1760 Stores I/O Module

The MIL-STD-1760 Stores I/O Module shall provide all the MIL-STD-1760 I/O interfaces to a stores station (video, audio, analog, discretes, and power). The module will have a PI-Bus interface, Test and Maintenance Bus interface, and two Video Data Distribution Network interfaces. The MIL-STD-1760 Stores I/O Module is shown in Figure 6.1.19-1.

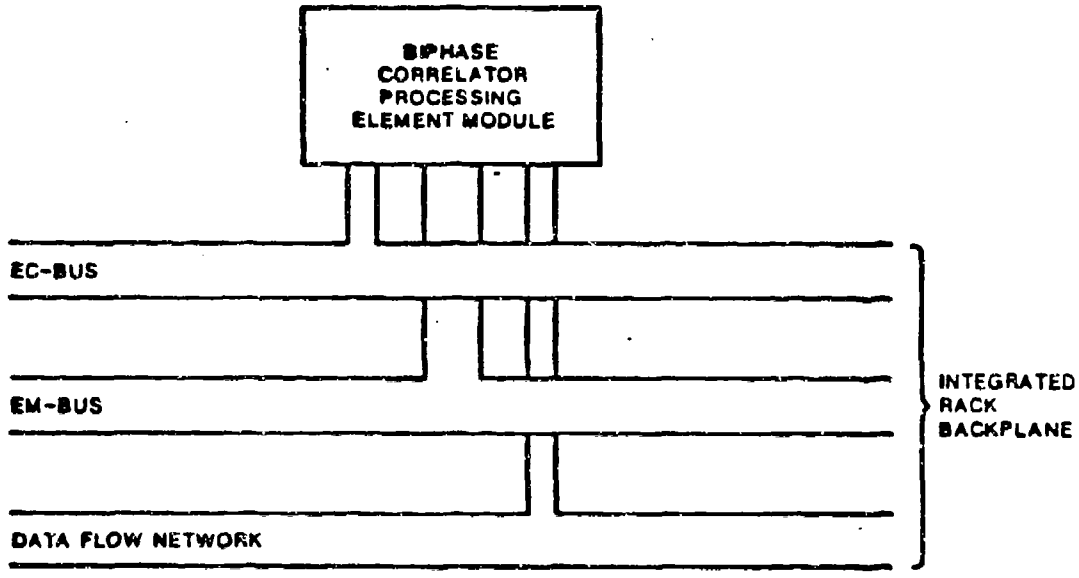


Figure 6.1.18-1 **BIPHASE CORRELATOR PROCESSING ELEMENT MODULE**

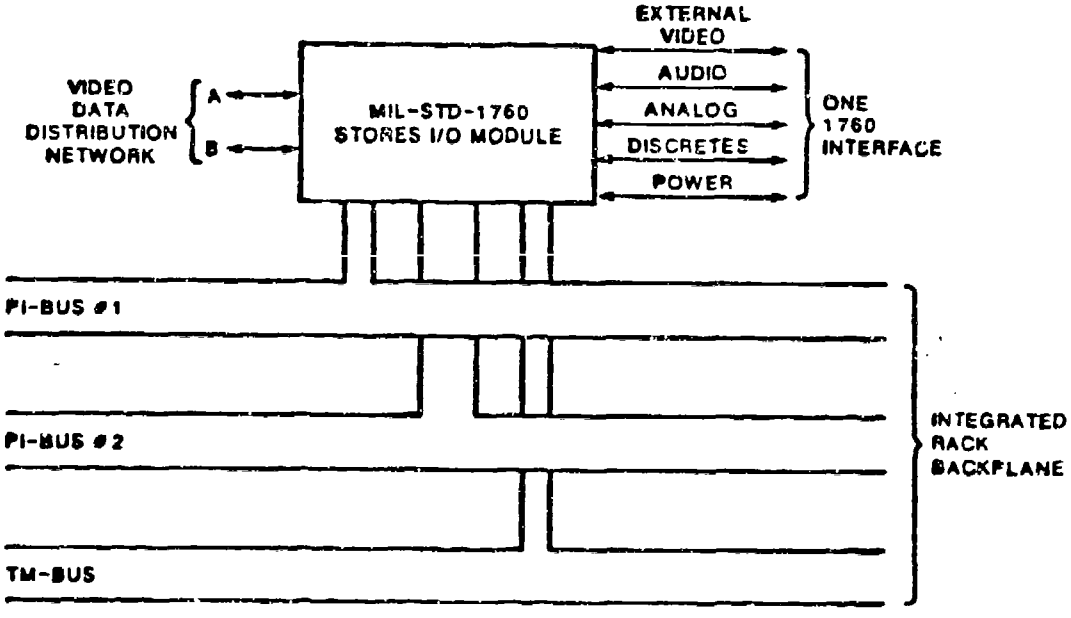


Figure 6.1.19-1 **MIL-STD-1760 STORES I/O MODULE**

6.1.20 Video Module

The Video Module shall be capable of extracting information from data memory and creating pictures for display. The module shall do character generation and shall be capable of outputting high resolution color video for display in the cockpit. The video module shall also be capable of accepting and processing video data received from the stores stations and providing this data to the video displays. The video memory module shall provide the ability to store many types of video data (monochrome, color, etc) based on multiples of the minimum number of TV lines. A 512 x 512 x 16 module for example could be configurable to permit:

- -512 x 512 x 12 (four bit planes for each primary color)
- -1024 x 512 x 4 (monochrome)
- -1024 x 1024 x 4 (monochrome)
- -Simultanous buffering for radar/EO/IRST
- -Backup for failed memory modules

The video module is a functional element in the Common Signal Processor and contains Element Control Bus, Element Maintenance Bus, and Data Flow Network interfaces. The Video Module is shown in Figure 6.1.20-1.

6.1.21 Display Module

The Display Module is the final destination of Video Data from the Video Distribution Network and is resident with the display device. It interfaces directly to the display electronics through the Data Flow Network. The Video Module also interfaces to the PI Bus for control and the Test and Maintenance Bus. The Display Module is shown in Fig. 6.1.21-1.

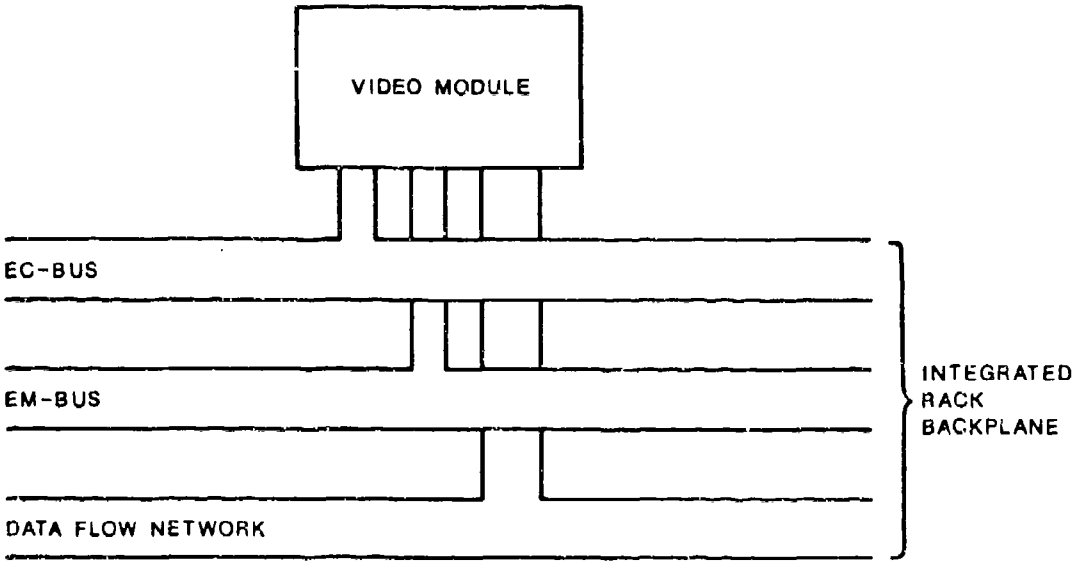


Figure 6.1.20-1 VIDEO MODULE

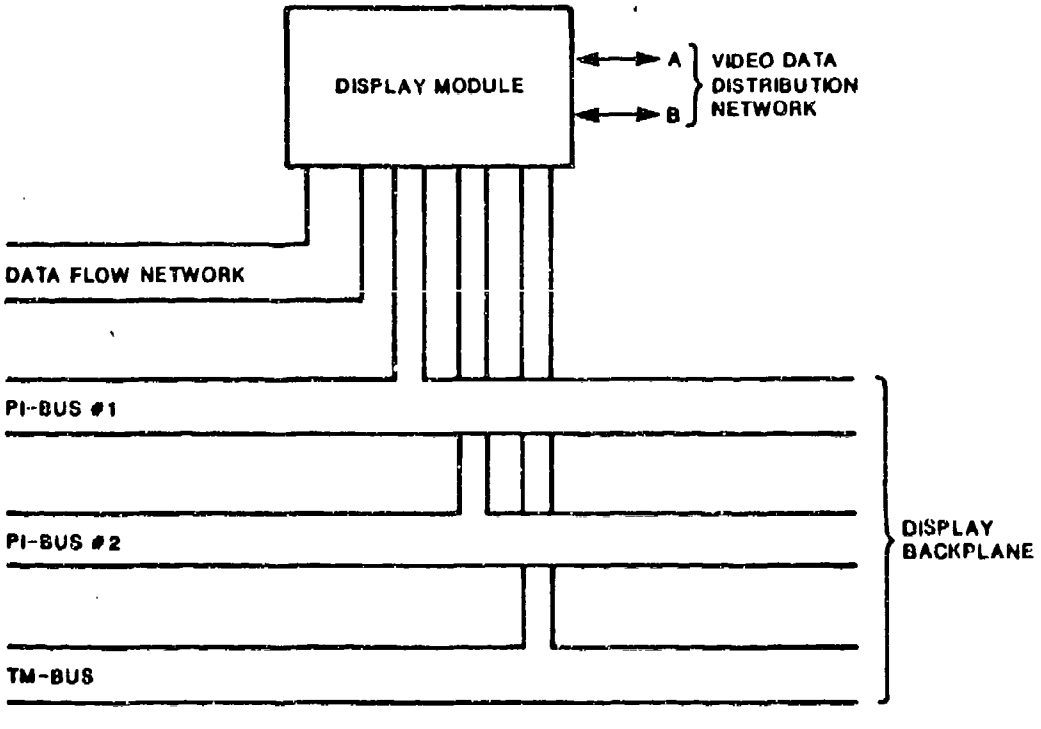


Figure 6.1.21-1 DISPLAY MODULE

6.1.22 Network Switch Module

The Network Switch Module is shown in Figure 6.1.22-1. It is used as shown in Figure 6.1.22-2 to provide the interconnect routing for the SDDN, DEN and VDDN. The NSM shall be capable of connecting any of the 500 MBPS fiber optic input channels to any of the 500 MBPS output channels under command of a local V1750A control processor via dual redundant PI-Bus interfaces. The prime data path is between NSM input channels and NSM output channels. Requests to configure the input to output connectivity are received on the Mission Avionics Multiplex Bus and passed to the control processor via the local PI-Bus. Switch test, status and reconfiguration commands are sent to the switch from the local controller via the local PI-Bus. Switch status is returned to the local controller via the local PI-Bus.

The NSM shall support non-blocking n X n interconnect between input and output channels at the maximum network data rate (500 MBPS per channel). The number of channels per NSM, the allocation of channels among sources and sinks, and the number of NSMs per network are system design dependent parameters.

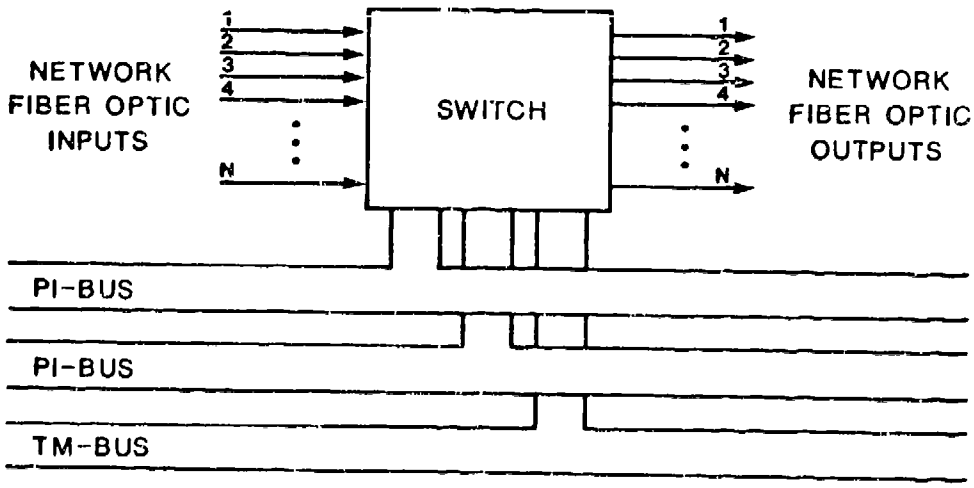


Figure 6.1.22-1 NETWORK SWITCH MODULE

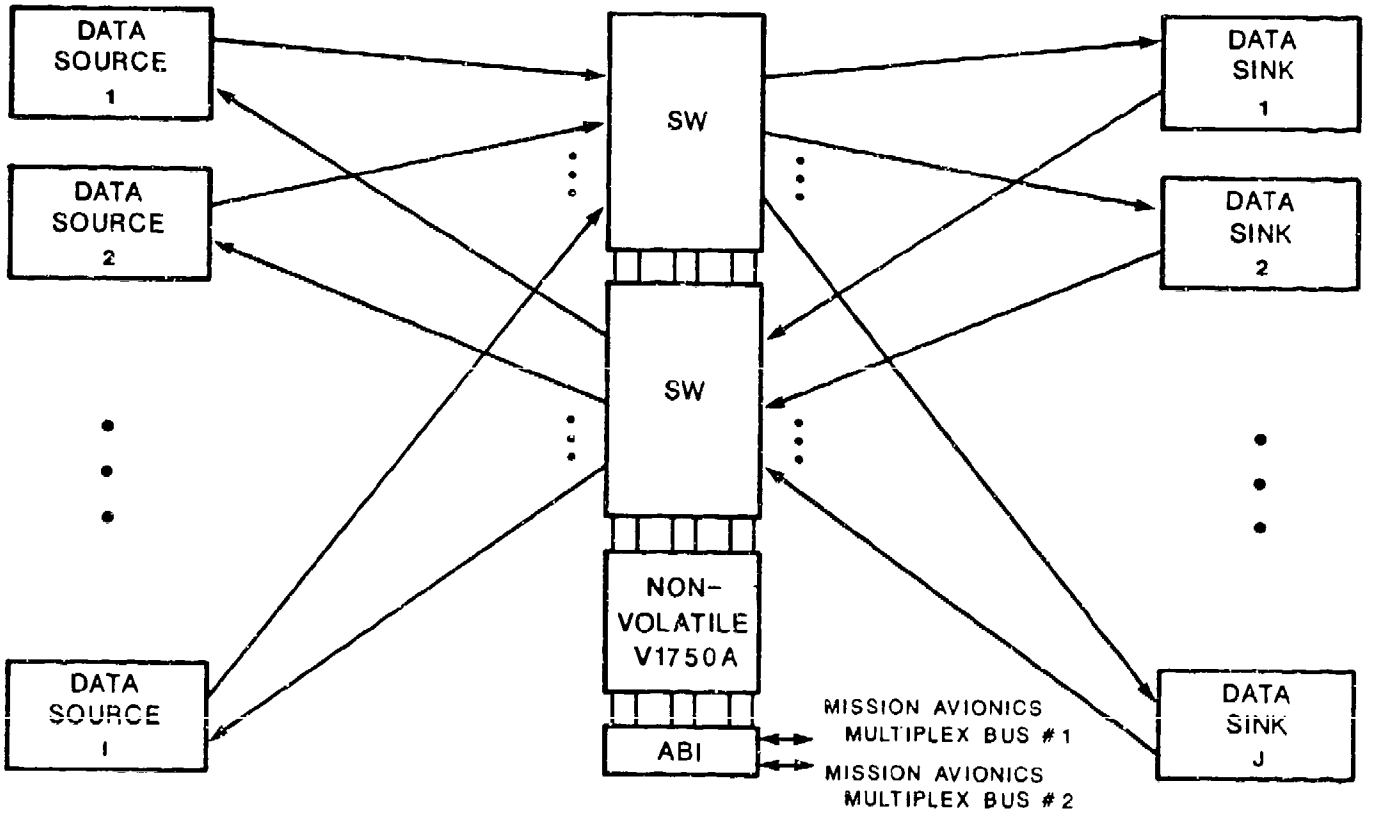


Figure 6.1.22-2 NETWORK SWITCH UTILIZATION

6.1.23 Key Generator Module (KGM)

The Key Generator Module (KGM) shall be a general purpose cryptographic module capable of encryption and decryption of all digital data and messages including IFF and voice signals. It shall provide COMSEC protection of digital data at all classification levels. The KGM shall also accept manual key load, fault alarm, tamper detect, and internal zeroize functions. It shall perform multi-mode operation at a data rate of 20 Mbps, with a maximum latency of 100 microseconds. The KGM shall have a crypto ignition key interface to allow the pilot to enter key code inputs for access to secure data. The KGM module is shown in Figure 6.1.23-1.

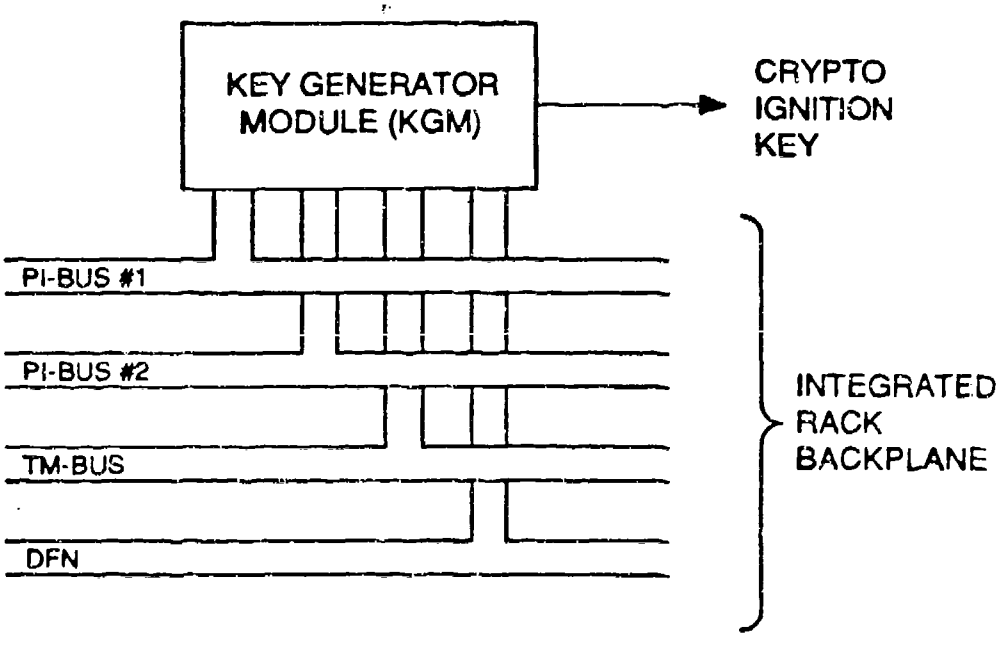


FIGURE 6.1.23-1 KEY GENERATOR MODULE

6.2 Packaging

The objective of avionics packaging is to develop a standardized set to allow interchangeability of equipment within an aircraft and also between different aircraft types which contain similar functional units. The packaging concept will provide:

- A set of line replaceable modules (LRM)
- A set of standardized avionics enclosures/racks.

The key design objectives and constraints include the following:

- Direct access to the LRMs
- Minimizing installed avionics weight and volume
- Mechanical simplicity
- Guards against inadvertent installation of an LRM into the wrong slot
- Effective thermal control of the equipment
- Improvements in reliability and maintainability
- Improved testability to allow fault isolation to LRM level
- Improved tolerance to battle damage
- Elimination of single point failure modes.

6.2.1 Equipment Enclosure

The equipment enclosures shall be modularized with standard dimensions to allow use of building block subassemblies. An integrated rack design compatible with two-level maintenance procedures shall be used.

Liquid cooling shall be the primary mechanism of heat dissipation for the equipment cooling system. The

enclosure interface with the equipment cooling system shall be specified in terms of heat dissipation requirements. Loss or reduction of cooling capacity shall not cause degradation in avionics performance below specified limits or damage to the LRM or enclosure within a given length of time. The enclosure shall be capable of operating in the applicable service condition of MIL-E-5400.

The enclosure and connector design shall incorporate means to exclude radiated or conducted EMI originating outside the rack. The avionics as installed shall meet the requirements of MIL-E-6051. The enclosure shall also comply with the applicable requirements of MIL-STD-461. EMI/EMP requirements shall be considered in material selection for the enclosure. All metal parts of the enclosure shall be maintained at airframe potential by application of suitable bonding and grounding techniques.

EMP hardening shall be applied to the enclosure. The following techniques shall be used: (1) reduction or elimination of collected energy by application of shielding and grounding and bonding; (2) reduction of incident energy on equipment using protection devices (filtering and amplitude limiting); (3) increasing failure threshold of circuits by component selection; (4) modifying circuit function (functional hardening) to reduce susceptibility, and (5) circumvention.

Electrical power shall be provided at the enclosure. Primary power shall be 3 phase 115 VAC, 400 Hz. Internal power supplies shall provide voltage at the working levels required. 28 VDC shall be provided if no-break power is required. Fault isolation and protection shall be provided within the enclosure.

6.2.2 Line Replaceable Module

Sets of interchangeable standard modules shall be used. The size of the modules shall be such as to allow replacement of a complete functional unit such as a 1750A Processing Module. The modules shall be designed to meet the existing specifications for integrated circuits, passive components, hardware (connector assemblies) and printed circuit cards. The primary mode of heat transfer shall be by conduction to slots in the enclosure. Alternate cooling methods such as heat pipes shall be used when the heat dissipation requirements exceed the maximum allowable for conduction cooled modules. The modules shall meet the applicable specifications for thermal, EMI/EMP and structural integrity. Standardized connectors with a discrete set of pin numbers shall be used. The connector design shall address wire wrap capability, multi-layer board design constraints and connector/module substrate interface compatibility. The module packaging shall allow use of leaded or leadless ceramic chip carriers or pin grid arrays. The substrate and chip carrier coefficient of thermal expansion shall be matched within specified limits. Appendix A contains detailed specifications for the 3/4 ATR modules to be used.

6.2.3 Damage Tolerance

The primary consideration shall be to reduce the probability of aircraft loss due to loss of flight essential avionics, with loss of mission avionics being given secondary consideration. The avionic elements shall be physically separated in different enclosures to minimize loss of any function due to battle damage. Separation shall also be provided within the enclosure of critical independent elements to limit single projectile damage effects.

6.3 Executive to Applications Interface

This section provides the definition of interfaces between application (mission) software and the executive software. The executive software is described in Section 3.7.11, 3.7.12 and 3.7.13. In this context, applications software is defined as all software not a part of the executive software. With the exception of inter- module interfaces, as discussed in Section 3.3.8.3, all interfaces made by an applications module are required to be with the executive. The executive to applications interfaces are, therefore, required to provide all capabilities necessary to the participation of a module in the execution of its parent function and by extension of the system. The executive interfaces described in this section provide for applications interfaces with the executive and for executive interfaces with applications tasks.

6.3.1 Applications Interface with the Executive

These interfaces provide access to the Kernel Executive. All applications tasks shall utilize these interfaces to the exclusion of any others for all classes of interface except inter-module.

6.3.2 Executive Interfaces with Applications Tasks

The executive interfaces with applications tasks primarily to perform task control operations resulting from applications to executive interfaces described above. Task execution, termination, cancellation, suspension, and restoration are the five primary items. Task priority and the event conditions associated with suspension are the primary terms defining task restoration. An indirect interface with applications tasks exists when the executive performs data base manipulation operations. These operations occur under direction of applications tasks. The Kernel Executive handles

all direct (i.e., task control) interfaces, while the Kernel Executive coordinates with ihe Distributed Executive to provide indirect (i.e., data base) interfaces.

6.3.3 Usage of Executive/Applications Interfaces

All applications tasks are required to utilize the applications to executive interfaces for all activities external to the specific task. The usage of executive to applications interfaces is restricted to the execution, and may be effected as required by the executive. The executive may, in the interest of efficiency and modularity, also utilize certain of the applications to executive interfaces; for example, a request by a task to invoke another task (the RUN applications to executive interface) will cause the executive, as a part of its processing, to enable the task for execution (an executive to application interface) as well as request to itself that an event honoring the occurrence be issued (the SIGNAL applications to executive interface).

Applications to executive interfaces will be implemented using packages (an Ada construct) to restrict the ability of the applications task to access information. The package utilized by the applications tasks will be a variation of a similar package utilized by the executive. The variation will consist of a restricted visible portion of the package. The amount of the package visible to the applications task will be that which is necessary and sufficient to permit the required task access. For example, an applications task which desires only to access data will utilize a package with a visible portion which permits access only to the data area and to the access routine. Since this is a subset of the package utilized by the executive, it tends to insure that the constructs and routines accessed by an applications task are actually supported by the executive.

During the period in which Ada is not available, interface constructs will be utilized during any necessary simulations as part of this effort. A language permitting a measure of this capability is MIL-STD-1598B JOVIAL, which permits the use of DEFINEs to implement the restricted interfaces.

6.4 Non-Homogeneous Processing Resources

The fully integrated architectural concept includes a set of common data and signal processing resources applied to meet the processing requirements of all of the sensors and subsystems. This set of processing resources is reconfigurable and supports the inclusion of spare elements in order to provide higher probability of availability/mission success and to support graceful degradation. The architecture must also support the integration of sensors/subsystems that are not fully integrated and the system implications of the various levels of sensor/subsystem integration. In addition, the architectural concept includes, as the normal case, identical data and signal processing resources. However, in a specific application it may not be feasible to design all processing elements to perform all jobs. Therefore, this section also addresses the issues of non-homogeneous processing resources.

6.4.1 Standalone Subsystem

In the case of a Standalone Subsystem, the subsystem consists of a collection of dedicated sensor elements and a set of dedicated signal and data processing resources (hardware and software). These processing resources are dedicated to only this one job, and cannot participate in system level resource sharing. This subsystem must be self-contained in that it may not be initially loaded by the core system, and it must internally contain any fault tolerant features that it requires to meet high availability/mission success goals. Self-contained functional subsystems of this nature shall interface

to either the Mission Avionics Multiplex Bus or the VMS Multiplex Bus, depending upon the flight critical nature of the function of the subsystem. In addition, if the subsystem generates or uses as an input a video signal, it shall interface to the system Video Data Distribution Network. These interfaces are shown in Figure 6.4.1-1. The subsystem shall possess well-defined message structures specifying the data that it shall receive from the multiplex bus, and specifying the data that it shall place onto the multiplex bus. The subsystem shall be capable of participating in the system level control procedures for operation of the multiplex bus structure, but shall act only as a remote device in terms of any higher level system control.

6.4.2 Shared Data Processing Subsystem

In the case of a subsystem which shares data processing, the subsystem consists of a collection of dedicated sensor elements and a set of dedicated signal processing resources (hardware and software). The data processing functions associated with the subsystem are partitioned onto parts of the array of system level data processing resources. These data processing resources may be reassigned to alternate functions in major mode changes or in failure recovery modes. The signal processing resources are dedicated to only this one job, and cannot participate in system level signal processor resource sharing. The signal processor resources in this subsystem must be self-contained in that they may not be initially loaded by the core system, and they must internally contain any fault tolerant features that are required to meet high availability/minimum success goals. Subsystems of this nature will interface to the Mission Avionics Multiplex Bus or the VMS Multiplex Bus, depending upon the flight critical nature of the function of the subsystem. If the sensor generates or uses as an input a video signal, it will interface to the system Video Data Distribution Network. These interfaces are shown in Figure 6.4.2-1. In addition, the data processing software to

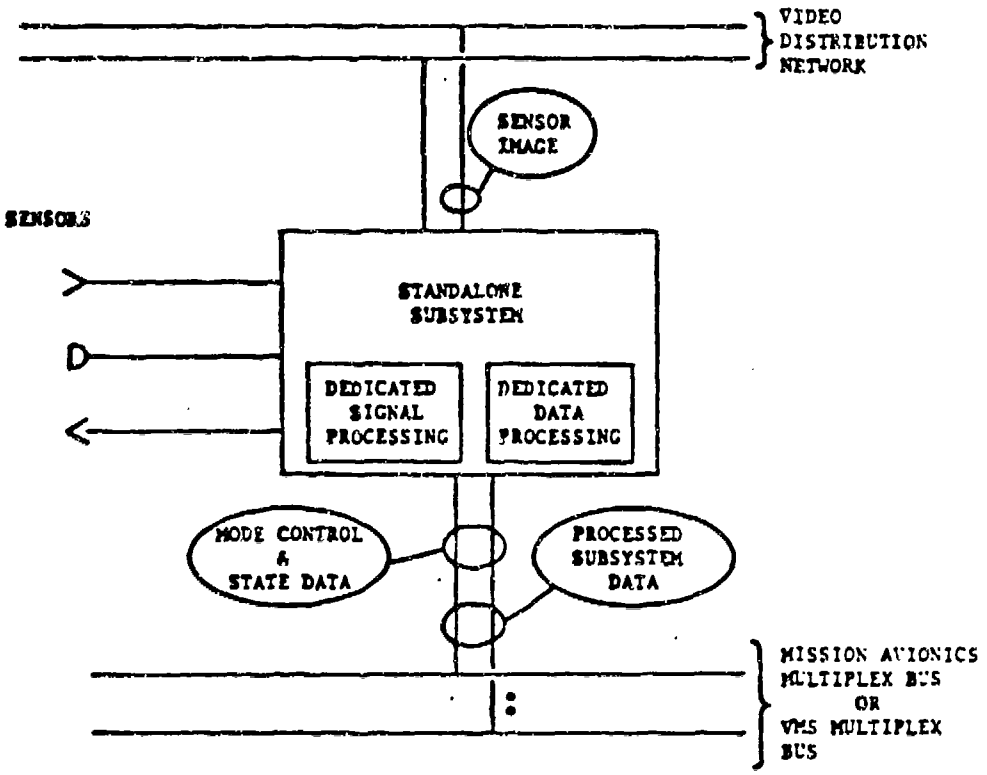


Figure 6.4.1-1 STANDALONE SUBSYSTEM INTERFACES

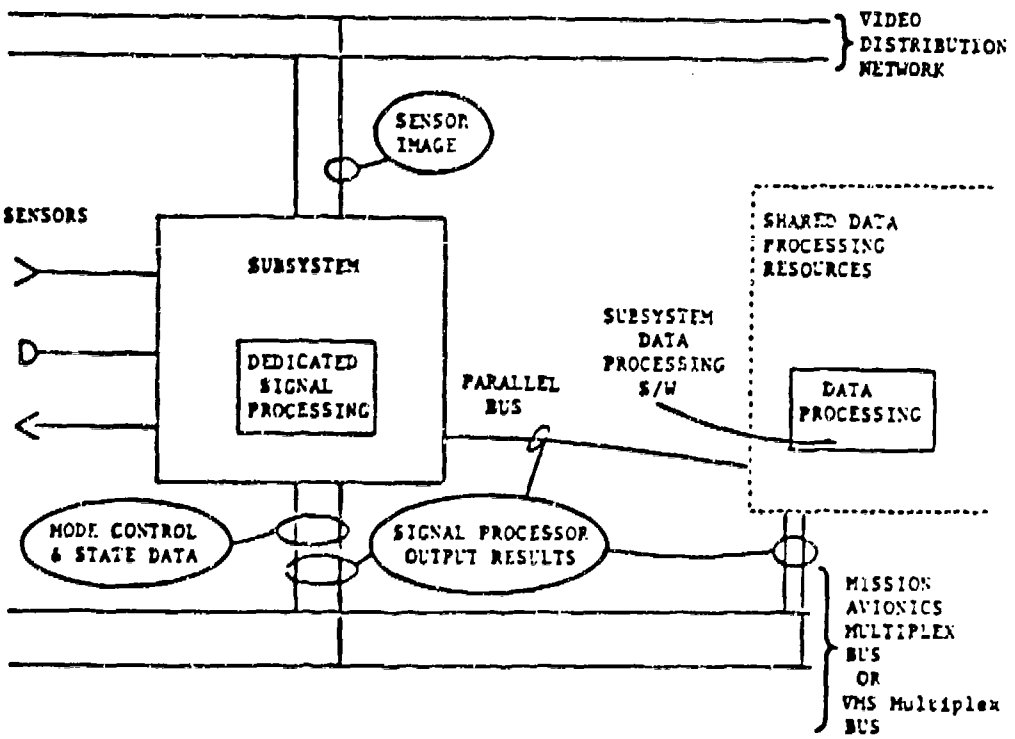


Figure 6.4.2-1 SHARED DATA PROCESSOR SUBSYSTEM INTERFACES

support this subsystem shall reside on some part of the system sharable data processing resources. Communication between the signal and data processors shall be over the Mission Avionics/VMS Multiplex Bus, as will communications between the data processing function and any users of these subsystem results. The subsystem shall possess well defined message structures specifying the data that it shall receive and transmit on the multiplex bus. The subsystem shall be capable of participating in the system level control procedures for operation of the multiplex bus structure, but shall act only as a remote device in terms of any higher level system control.

6.4.3 Shared Data and Signal Processing Subsystem

In the case of a subsystem that shares signal and data processing the subsystem shall consist of a collection of dedicated sensor elements and a set of dedicated sensor pre-processing hardware custom to that subsystem. The signal processing functions associated with the subsystem are partitioned onto parts of the array of system level signal processors, and the data processing functions are partitioned onto parts of the array of system level data processors. Subsystems of this nature shall interface to the Data Distribution Network to provide their data to the signal processing function, and shall receive control, data and timing standard from the Sensor Control Network. These interfaces are shown in Figure 6.4.3-1. Communications between the shared signal and data processors shall be on the Mission Avionics Multiplex Bus. The subsystem shall possess well-defined message structures for its data sent to the Signal Processing, its data received from the Signal Processing, and the communications between the shared Signal and Data Processing Resources. The subsystem shall be capable of participating in the communications protocols for the Data Distribution Network and the Sensor Control Network.

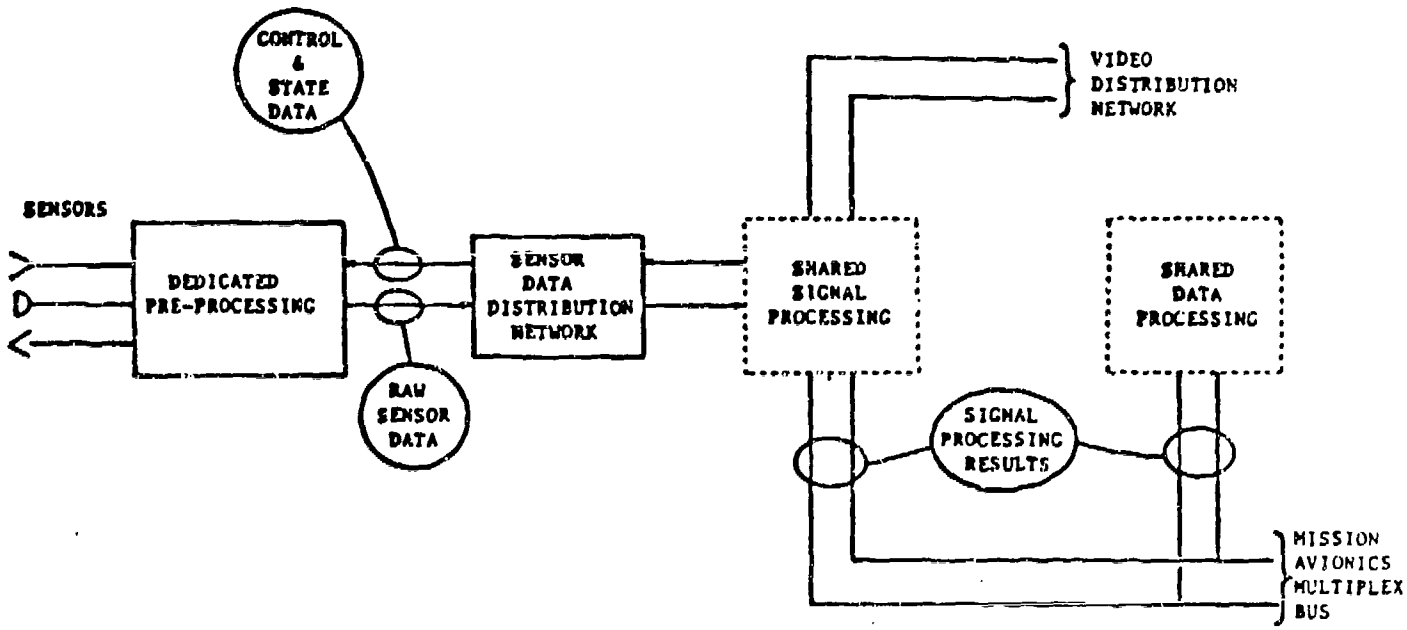


Figure 6.4.3-1 SHARED DATA & SIGNAL PROCESSING SUBSYSTEM INTERFACES

6.4.4 Incomplete Sensor Interconnectivity

The fully integrated system concept assumes that a single Sensor Data Distribution Network fully connects all dedicated Sensor preprocessors to the full array of signal processors. In specific implementations it may be desirable to interconnect these elements. This case is shown in Figure 6.4.4-1. In this case, additional restrictions on the allowable assignments of functions onto signal processors must be taken into account in the System Executive Function.

6.4.5 Non-Homogeneous Data Processors

The basic Architectural Concept assumes as a default case that all data processing elements are identically configured, and are the lowest elements within which resource allocations can occur. However, in specific system implementations it may be necessary to include data processing elements which possess additional resources (memory, throughput, spare resources). As a result, special consideration shall have to be made at the system level with respect to what functions can be assigned to these non-standard data processors, and to how those functions can be supported in failure modes.

6.4.6 Non-Homogeneous Signal Processors

The basic Architectural Concept assumes a bank of identically configured Signal Processors, any of which can be assigned to any job. However, in a specific system where one job far exceeds the requirements of all other jobs, it may be necessary to consider more than one configuration of Signal Processors. As a result, additional constraints shall be introduced into the system control process on assignment of functions to Signal Processors in both normal and failure modes.

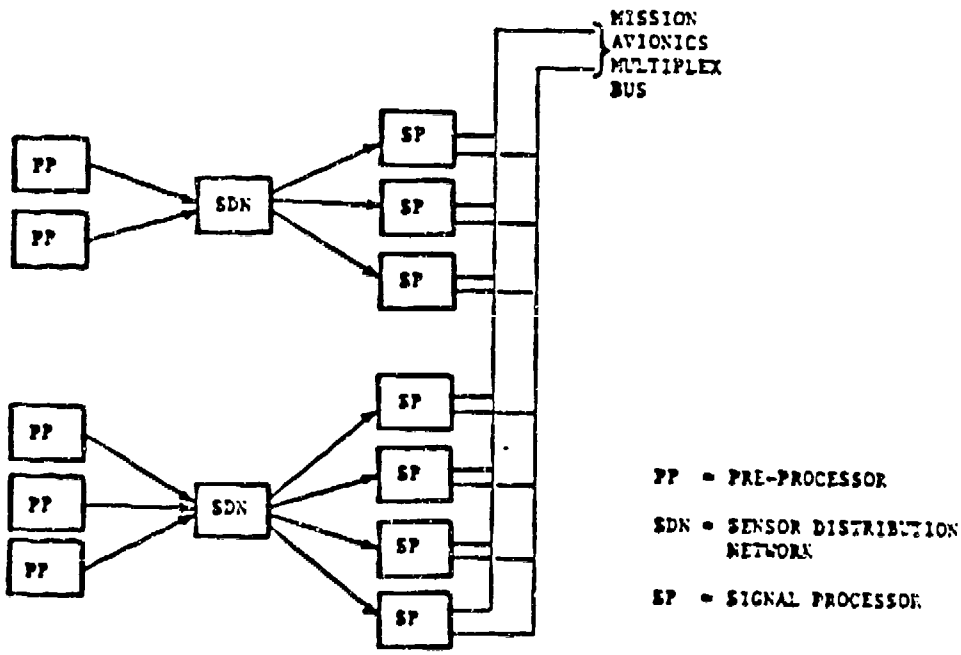


Figure 6.4.4-1 INCOMPLETE SENSOR INTERCONNECTIVITY

6.5 ACRONYMS

ADM	Advanced Development Model
APSE	Ada Programming Support Environment
ATR	Air Transportable Rack
BCPE	Biphase Correlator Processing Element
BMM	Bulk Memory Module
CNI	Communications Navigation Identification
COMSEC	Communications Security
CSP	Common Signal Processor
DAIS	Digital Avionics Information System
DCI	Dataflow Control Interface
DEN	Data Exchange Network
DFN	Data Flow Network
ECB	Element Control Bus
EDC	Error Detecting and Correcting
EMB	Element Maintenance Bus
ESU	Element Supervisor Unit
EW	Electronic Warfare
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FPPE	Floating Point Processing Element
FSED	Full Scale Engineering Development
GM	Global Memory
GPPE	General Purpose Processing Element
GPS	Global Positioning System
H/W	Hardware
HZ	Hertz (Cycles Per Second)
IDS	Interface Design Specification
IIR	Infinite Impulse Response
INS	Inertial Navigation System
I/O	Input/Output
ISA	Instruction Set Architecture
JTIDS	Joint Tactical Information Distribution System
KGM	Key Generator Module
LCM	Load Control Module
LRM	Line Replaceable Module
MAMB	Mission Avionics Multiplex Bus
MBPS	Million Bits Per Second
MHZ	Million Cycles Per Second
MIPS	Million Instructions Per Second
MOPS	Million Operations Per Second
MSEC	Milli-Second
NED	Nuclear Event Detection
NSM	Network Switch Module
NVBMM	Non-volatile Bulk Memory Module
OFP	Operational Flight Program

PDL	Program Design Language
PE	Processing Element
PI-BUS	Parallel Inter-Module Communication Bus
PVI	Pilot Vehicle Interface
RISA	Reduced Instruction Set Architecture
ROM	Read Only Memory
RT	Remote Terminal
SDDN	Sensor Data Distribution Network
SDU	Secure Data Unit
SEPE	Sort Enhanced Processing Element
SI	Sensor Interface
SMS	Stores Management System
SP	Signal Processor
S/W	Software
TCG	Timing and Control Generator
TF/TA/OA	Terrain Following/Terrain Avoidance/ Obstacle Avoidance
TM-BUS	Test and Maintenance Bus
TRANSEC	Transmission Security
TTL	Transistor-Transistor Logic
UCIF	User Console Interface
URT	Universal Remote Terminal
V1750A	VHSIC MIL-STD-1750A
VDDN	Video Data Distribution Network
VHSIC	Very High Speed Integrated Circuit
VMS	Vehicle Management System
VPE	Vector Processing Element

APPENDIX A

GENERAL SPECIFICATION FOR 3/4 ATR MODULE

1.0 SCOPE.

1.1 Purpose. The purpose of this document is to establish the general design requirements for the 3/4 ATR module.

1.2 Classification. Standard electronic modules shall be of the following classes as specified in detail specifications:

Class II - For utilization where stringent environmental requirements are imposed.

Class IV - For utilization where class II modules may be exposed to radiation.

2.0 REFERENCED DOCUMENTS.

2.1 Issues of documents. The following documents, of the issue in effect on date of invitation for bids or request for proposal, form a part of this specification to the extent specified herein.

SPECIFICATIONS

MILITARY

MIL-A-8625 - Anodic Coatings, for Aluminum and Aluminum Alloys.

MIL-E-16400 - Electronic, Interior Communication and Navigation Equipment, Naval Ship and Shore, General Specification for.

MIL-S-19500 - Semiconductor Device, General Specification for.

MIL-C-26074 - Coating, Electroless Nickel, Requirements for.

MIL-C-28754 - Connector, Electrical, Modular, and Component Parts, General Specification for.

MIL-M-28787 - Modules, Standard Electronic, General Specification for.

MIL-M-38510 - Microcircuit, General Specification for.

MIL-C-39003 - Capacitors, Fixed, Electrolyte (Solid Electrolyte), Tantalum, Established Reliability, General Specification for.

MIL-P-50884	- Printed Wiring, Flexible, General Specification for.
MIL-S-83490	- Specifications, Types and Forms.

STANDARDS

MIL-STD-12	- Abbreviations for Use on Drawings, Specifications, Standards and in Technical Documents.
DOD-STD-100	- Engineering Drawing Practices.
MIL-STD-129	- Marking for Shipment and Storage.
MIL-STD-130	- Identification Marking of U.S. Military Property.
MIL-STD-202	- Test Methods for Electronic and Electrical Component Parts.
MIL-STD-242	- Electronic Equipment Parts Selected Standards (Part 1 through Part 8).
MIL-STD-275	- Printed Wiring for Electronic Equipment.
MIL-STD-454	- Standard General Requirements for Electronic Equipment.
MIL-STD-810	- Environmental Test Methods.
MIL-STD-883	- Test Methods and Procedures for Microelectronics.
MIL-STD-961	- Outline of Forms and Instructi ns for the Preparation of Specifications nd Associated Documents.
MIL-STD-1285	- Marking of Electrical and Electronic Parts.
MIL-STD-1378	- Requirements for Employing Standard Electronic Modules.
MIL-STD-1389	- Design Requirements for Standard Electronic Modules.

(Copies of specifications, standards, handbooks, drawings, and publications required by manufacturers in connection with specific acquisition functions should be obtained from the contracting activity or as directed by the contracting officer.)

2.2 Other publications. The following documents form a part of this standard to the extent specified herein. Unless otherwise indicated, the issue in effect on the date of invitation for bids or request for proposal shall apply.

Handbook H4-2 - Federal Supply Code for Manufacturers (United States and Canada) Code to Name.

Handbook H6 - Federal Item Name Directory.

(Applications for copies should be addressed to the Defense Logistics Agency, Defense Logistics Service Center, Battle Creek, Michigan 49016.)

BUMED INSTRUCTION 6270.3- Personnel Exposure Limits Values for Health Hazardous Air Contaminants.

(Applications for copies should be addressed to Naval District Washington, Supply and Fiscal Department (Code 514.3), Washington Navy Yard, Washington, DC 20390.)

2.3 Order of precedence. In the event of conflict the requirements specified in the contract, the detail specification, MIL-M-28787, this specification, and the documents referenced herein shall govern in that order.

3.0 REQUIREMENTS.

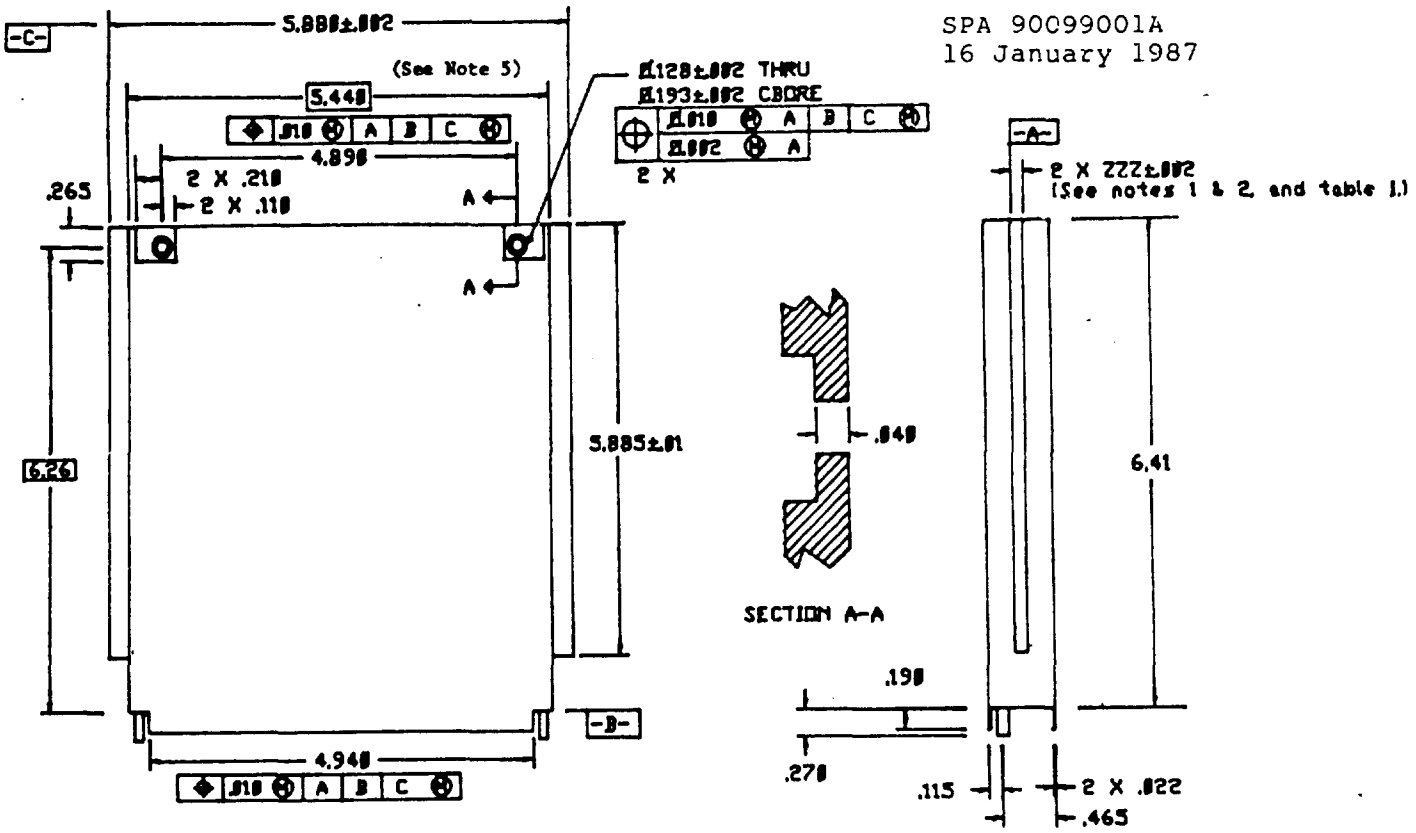
3.1 General requirements

3.1.1 Specifications and standards. All modules shall be in accordance with the requirements of the detail specification, MIL-M-28787, this standard, and MIL-STD-1378. Exceptions and alternates or equivalent materials, parts, processes, documentation, and so forth, shall be approved prior to their use in the design of a module.

3.1.2 Mechanical configuration requirements. The basic mechanical configuration of the module and connectors shall be as shown on figure 1 with incremental growth in thickness specified in table I.

3.1.3 Electrical configuration requirements. Electrical function and pin assignments shall be in accordance with table II. The maximum allowable current for each contact pin shall be 3 amperes dc or ac rms.

3.1.3.1 Preassigned dedicated pins: (see table II). Pins assigned functions which are not marked with an asterisk (* = optional) shall be considered preassigned and dedicated. Such pins shall be used for their preassigned function and used in accordance with the rules stated below or left unused.



INCHES	MM	INCHES	MM	INCHES	MM
.001	0.03	.025	0.64	.193	4.90
.002	0.05	.040	1.02	.210	5.33
.003	0.08	.04	1.0	.265	6.73
.004	0.10	.050	1.27	.270	6.86
.005	0.12	.0625	1.59	.465	11.81
.006	0.15	.100	2.54	4.890	124.21
.007	0.18	.110	2.79	4.940	125.48
.010	0.25	.115	2.92	5.440	138.18
.01	0.3	.125	3.18	5.880	149.352
.020	0.51	.128	3.25	5.885	149.48
.022	0.56	.19	4.8	6.26	159.0
				6.41	162.8

UNLESS OTHERWISE SPECIFIED TOLERANCES ARE:
3 place decimals + .005
2 place decimals + .01

NOTES:

1. Rib feature shall be contained within a .055 inch (1.40mm) maximum zone for the .050 inch ribs and a .130 (3.30mm) maximum zone for the .125 inch ribs.

2. Rib surface finish of 25 microinch or better.

3. Dimensions are in inches.

4. Metric equivalents are given for general information only.

5. Maximum Component area width of 5.44 inches allows minimum rib width of .22 inches. Maximum rib width is .30 inches which allows component area width minimum of 5.28 inches.

FIGURE 1. Module configuration.

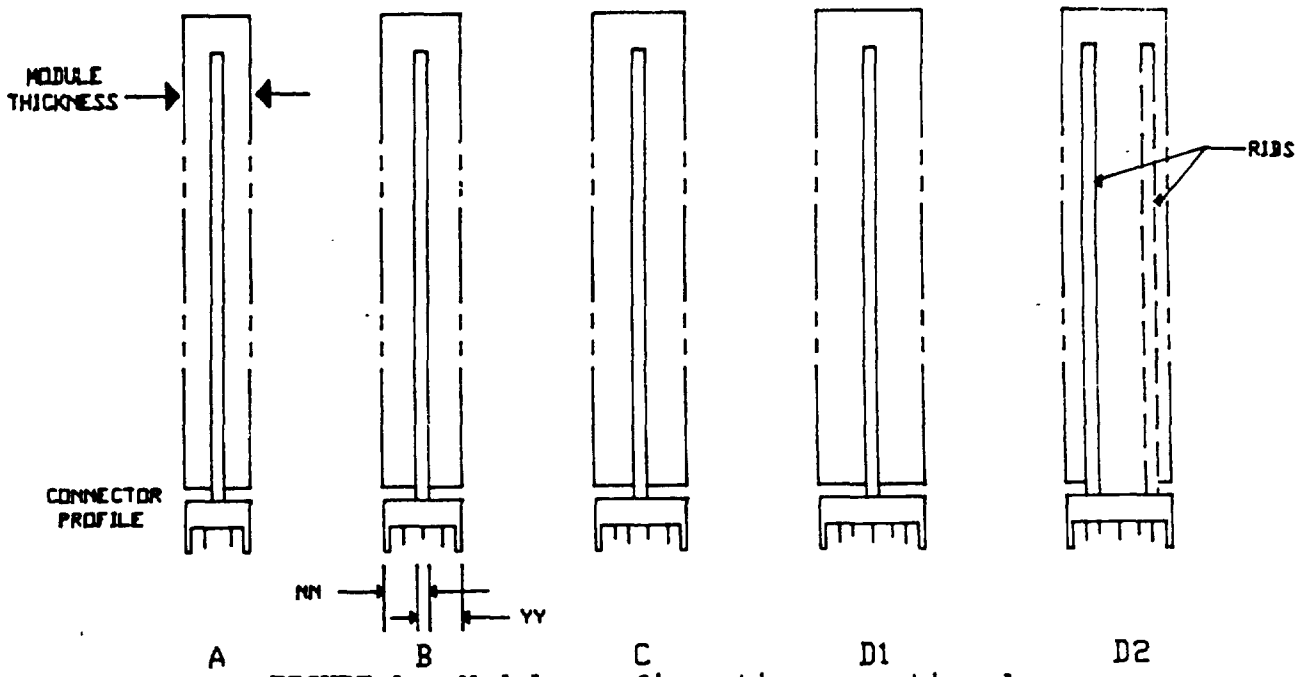


FIGURE 1. Module configuration - continued.

	A	B	C	D1	D2
Module pitch 1/	.3	.4	.5	.6	.6 2/
	(8)	(10)	(13)	(15)	(15)
Module thickness	.280	.380	.480	.580	.580
	(7.11)	(9.65)	(12.19)	(14.73)	(14.73)
Maximum number pins	100	150	200	250	250
Rib thickness (ZZZ)	.050	.125	.125	.125	XXX 3/
	(1.27)	(3.18)	(3.18)	(3.18)	
Dimension NN (max)	.152	.202	.252	.152	.302
4/	(3.86)	(5.13)	(6.40)	(3.86)	(7.67)
Dimension YY (max)	.152	.202	.252	.452	.302
4/	(3.86)	(5.13)	(6.40)	(11.48)	(7.67)

* Notes:

1/ Pitch refers to the distance between module centerlines for system packaging purposes.

2/ The .6 inch (15mm) pitch module configuration can increase in thickness in .1 inch (2.5 mm) increments with no increase in contact pin count.

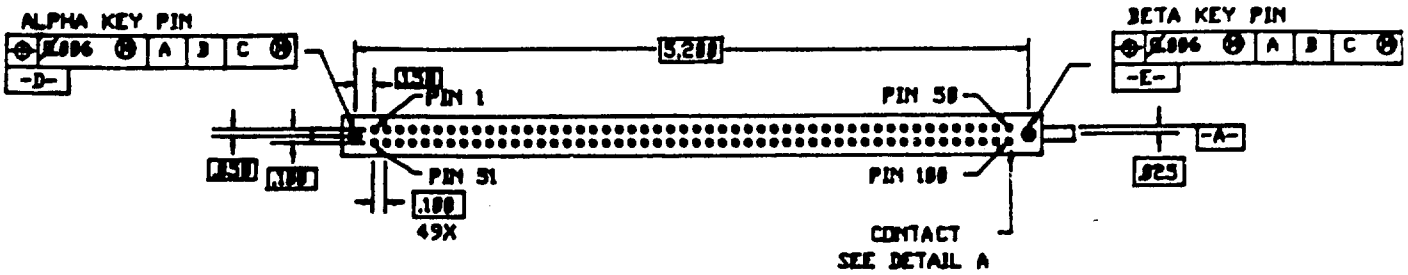
3/ Module configuration D2 can have either one rib at .125 inches (3.18mm) thick or two ribs at .050 inches (1.27mm) thick each.

4/ The dimensions are from the center of the two basic guide rib profiles to locate connector lateral extreme displacement.

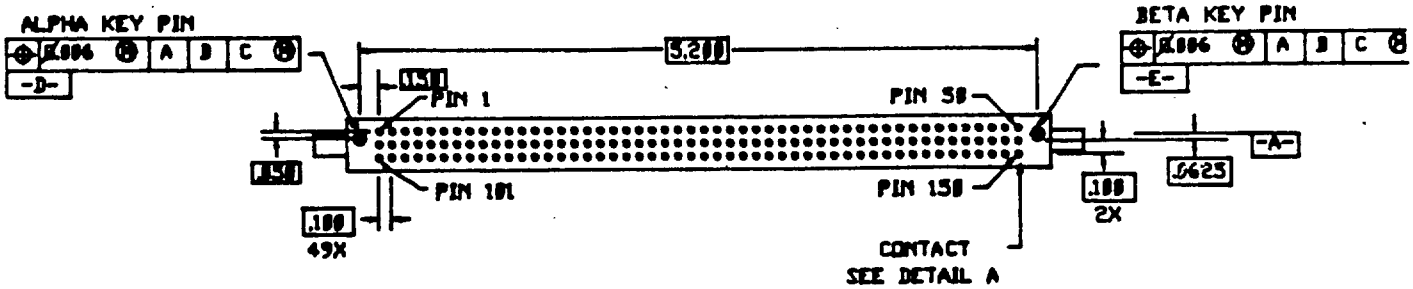
5/ Dimensions are in inches.

6/ Metric equivalents are given for general information only.

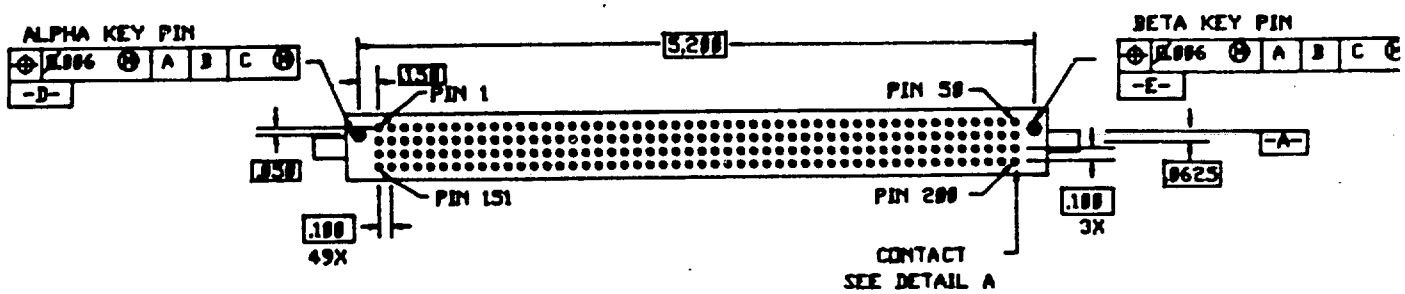
TABLE I. Module growth.



100 PIN CONNECTOR

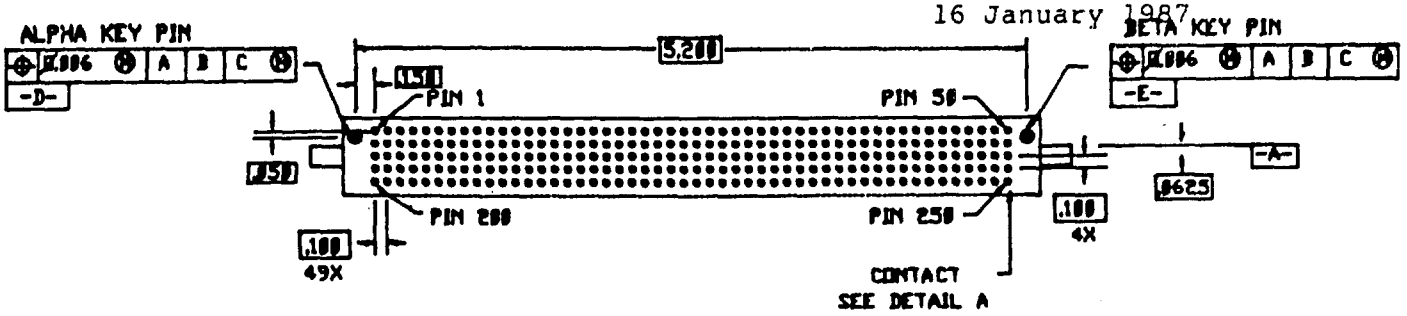


150 PIN CONNECTOR

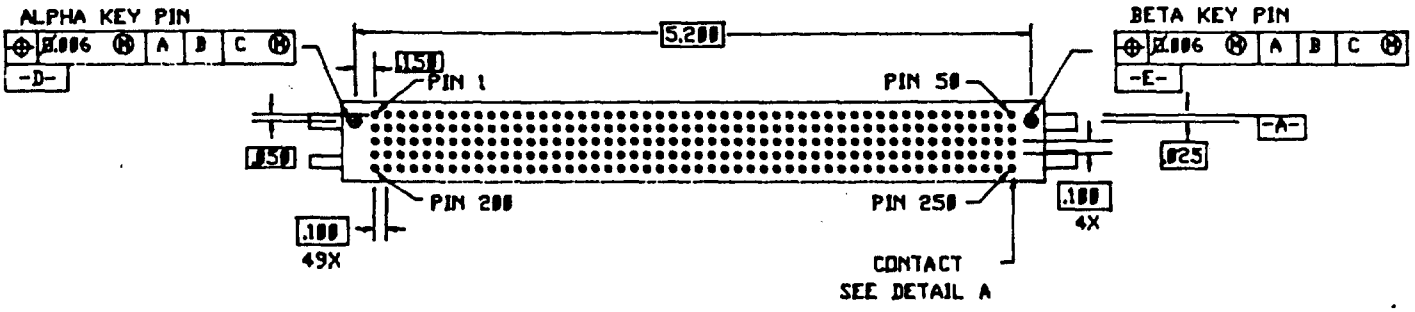


200 PIN CONNECTOR

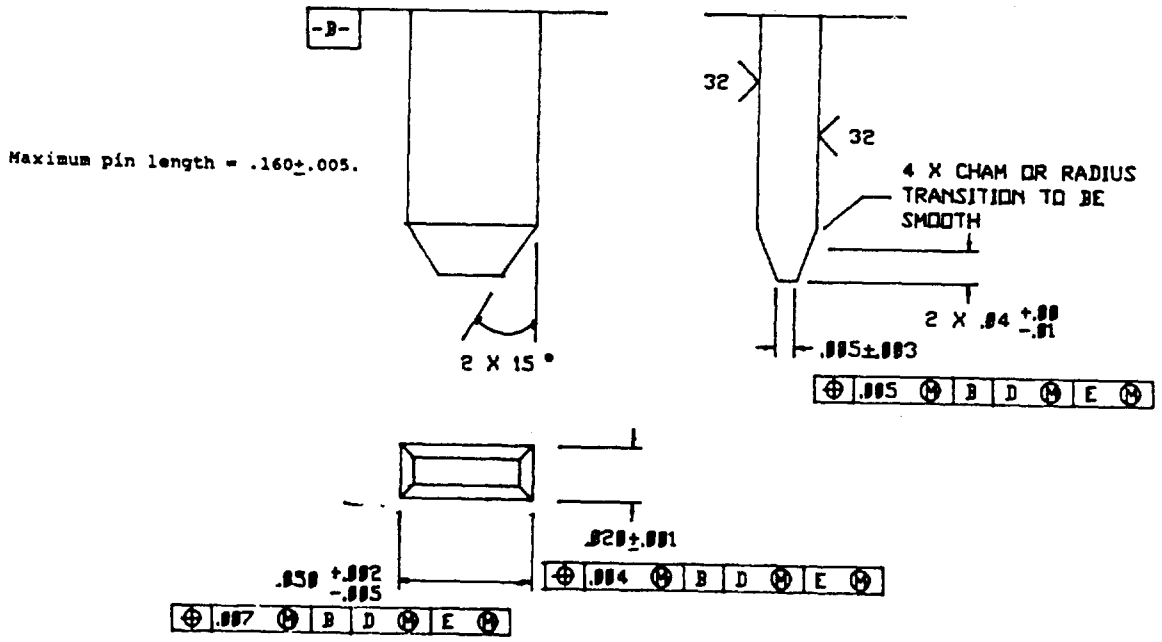
FIGURE 1. Module configuration - continued.



250 PIN CONNECTOR - CONFIGURATION D1



250 PIN CONNECTOR - CONFIGURATION D2



DETAIL A

FIGURE 1. Module configuration - continued.

	Row A		Row B		Row C		Row D		Row E
1.	+5 V	51.	+5 V*	101.		151.	+5 V*	201.	+5 V
2.		52.		102.		152.		202.	
3.	+V*	53.	+V*	103.		153.	+V*	203.	+V*
4.		54.		104.		154.		204.	
5.	GND*	55.	GND*	105.		155.	GND*	205.	GND*
6.		56.		106.		156.		206.	
7.		57.		107.		157.		207.	
8.		58.		108.		158.		208.	
9.		59.		109.		159.		209.	
10.	GND	60.	GND*	110.		160.	GND*	210.	GND
11.	Frame GND	61.		111.		161.		211.	
12.		62.		112.		162.		212.	
13.		63.		113.		163.		213.	
14.		64.		114.		164.		214.	
15.	Analog GND	65.	Analog GND*	115.		165.	Analog GND*	215.	Analog GND*
16.		66.		116.		166.		216.	
17.		67.		117.		167.		217.	
18.	+5 V*	68.	+5 V*	118.		168.	+5 V*	218.	+5 V*
19.		69.		119.		169.		219.	
20.	-V dc(1)*	70.	-V dc(1)*	120.		170.	-V dc(1)*	220.	-V dc(1)*
21.		71.		121.		171.		221.	
22.		72.		122.		172.		222.	
23.		73.		123.		173.		223.	
24.		74.		124.		174.		224.	
25.	GND	75.	GND*	125.		175.	GND*	225.	GND
26.		76.		126.		176.		226.	
27.		77.		127.		177.		227.	
28.		78.		128.		178.		228.	
29.		79.		129.		179.		229.	
30.		80.		130.		180.		230.	
31.	+5 V*	81.	+5 V*	131.		181.	+5 V*	231.	+5 V*
32.		82.		132.		182.		232.	
33.	+V	83.	+V*	133.		183.	+V*	233.	+V
34.		84.		134.		184.		234.	
35.	GND	85.	GND*	135.		185.	GND*	235.	GND
36.		86.		136.		186.		236.	
37.		87.		137.		187.		237.	
38.	-V dc(2)	88.	-V dc(2)*	138.		188.	-V dc(2)*	238.	-V dc(2)
39.		89.		139.		189.		239.	
40.	GND	90.	GND*	140.		190.	GND*	240.	GND
41.		91.		141.		191.		241.	CONN GRD (Not
42.		92.		142.		192.		242.	applicable
43.		93.		143.		193.		243.	for molded
44.		94.		144.		194.		244.	connectors)
45.	GND*	95.	GND*	145.		195.	GND*	245.	GND*
46.		96.		146.		196.		246.	
47.		97.		147.		197.		247.	
48.		98.		148.		198.		248.	
49.		99.		149.		199.		249.	
50.	+5 V	100.	+5 V*	150.		200.	+5 V*	250.	+5 V*

* Means optional.

TABLE II. Pin function (50-250 pins).

3.1.3.2 Optional pin assignment: (see table II). Pins assigned functions marked with an asterisk (*) shall be considered optional pins. They shall be used for their preassigned function or in accordance with the rules stated below.

a. Functions shall appear on dedicated pins before they appear on optional or unassigned pins.

b. Functions shall appear on optional pins before they appear on unassigned pins.

c. Fixed voltages of different potentials shall not be assigned to adjacent pins.

d. Any unused pins shall be isolated (not connected to any other used or unused pins).

e. Pins assigned as dedicated or optional pins may be considered as unassigned only after all the following conditions exist:

(1) The pin is not required for the preassigned function.

(2) There is a lack of pin availability.

(3) All other pin assignment requirements are met.

f. Optional pin locations shall be used for nonassigned functions before dedicated pins are used for nonassigned functions.

g. The analog ground shall only be used when two types of grounds are required. These two types of ground pins shall not be connected together internal to the module. Power ground shall be used if only one type of ground is required.

3.1.3.3 Connector pin assignments. The use of connector pin assignments of table II shall be as follows:

a. For a 100 pin input/output connector use rows A and E. Pins are to be numbered consecutively.

b. For a 150 pin input/output connector use rows A, B, and E. Pins are to be numbered consecutively.

c. For a 200 pin input/output connector use rows A, B, D, and E. Pins are to be numbered consecutively.

d. For a 250 pin input/output connector use rows A through E.

e. Figure 1 and Table I and II demonstrate configurations for 250 pins maximum. For a connector with more than 250 pins add a second row C for 300 pins and a third row C for 350 pins.

3.1.4 Thermal requirements. All modules shall be designed to be cooled through the ribs and shall be capable of being cooled by the rib with no other heat loss.

3.1.5 Environmental requirements. The following environmental requirements apply except as modified in 3.2.7.

3.1.5.1 Operating environmental requirements. Modules shall withstand without damage, the operating environmental requirements of MIL-M-28787.

3.1.5.2 Nonoperating environmental requirements. Modules shall withstand, without damage, the nonoperating environmental requirements of MIL-M-28787.

3.1.5.3 Hydrogen atmosphere. Modules using metal oxide thick film resistors shall be capable of passing the hydrogen atmosphere test specified in MIL-M-28787.

3.2 Detailed requirements.

3.2.1 Module construction. Modules shall conform to the design, construction, and physical dimensions specified herein and in MIL-M-28787. Modules of a given key code shall be mechanically and electricall interchangeable regardless of the system in which they are used when operated within the required module design limits.

3.2.1.1 Configuration. The basic module size has a span of 5.88 inches (149.4 mm) maximum, a thickness of 280 inches (7.11 mm) maximum, and is 6.68 inches (169.7 mm) maximum in total height. Modules may increase in thickness in accordance with table I. Dimensions and tolerances shall be as specified on figure 1.

3.2.1.2 Circuitry. The module shall have a minimum clearance of .015 inch (0.38 mm) around all edges of the substrate or printed-wiring board. The printed-wiring board shall be further reduced to allow for insertion of module extractor and prevent component damage during module extraction.

3.2.1.3 Module depth. The only parts of the module that shall extend below the interface plane are the keying pins, contact pins, and pin shields unless otherwise specified in the detailed module specification.

3.2.1.4 Module frame. The module frame shall include module rib structures and extraction capability. The module frame may also include protective covers and thermal clamps mounted to the rib structures. A module with thermal clamps mounted to it may violate the module envelope shown on figure 1. However, the clamps must be removable and the module without the thermal clamps mounted to it must fit within the envelope. The covers shall assist in EMI and CBR protection, and shall not violate the module envelope.

3.2.1.4.1 Module rib structure. The basic module configuration shall have a minimum of two ribs: one located at the alpha end and one located at the beta end of the module. The module ribs shall perform the following functions:

a. Alignment during insertion or extraction.

b. Retention.

c. Cooling.

The rib configuration and location is shown on figure 1. Modules of .3, .4 and .5 inch pitch shall have two ribs; one located at the alpha end and one located at the beta end of the module. These ribs shall be located as indicated in figure 1 and table I. Modules of .6 inch pitch shall have either two or four ribs. If four ribs are utilized, two shall be located at the alpha end and two shall be located at the beta end of the module. These ribs shall be located as indicated in figure 1 and table I. If only one rib is used on each end, it shall be located as indicated in figure 1 and table I. All ribs shall have a thickness as shown on table I.

3.2.1.4.1.1 Rib strength. The individual module ribs shall withstand a torque of 10 inch-pounds (1.13 newton-meters) minimum maintained for 10 to 15 seconds. There shall be no detrimental effect to the mechanical integrity of the module.

3.2.1.4.2 Module extractor interface. Modules shall either have two extractor holes located as shown on figure 1 or an alternate means of insertion/extraction. The latter may violate the module envelope. If extractor holes are present, modules having a thickness of 2.090 inches (53.09mm) or greater shall have two sets of extractor holes. The second set shall be located within the last .3 inch (8 mm) of the module and meet the location requirements shown on figure 1.

3.2.1.5 Module header structure. The module header shall perform the following functions:

a. Module identification.

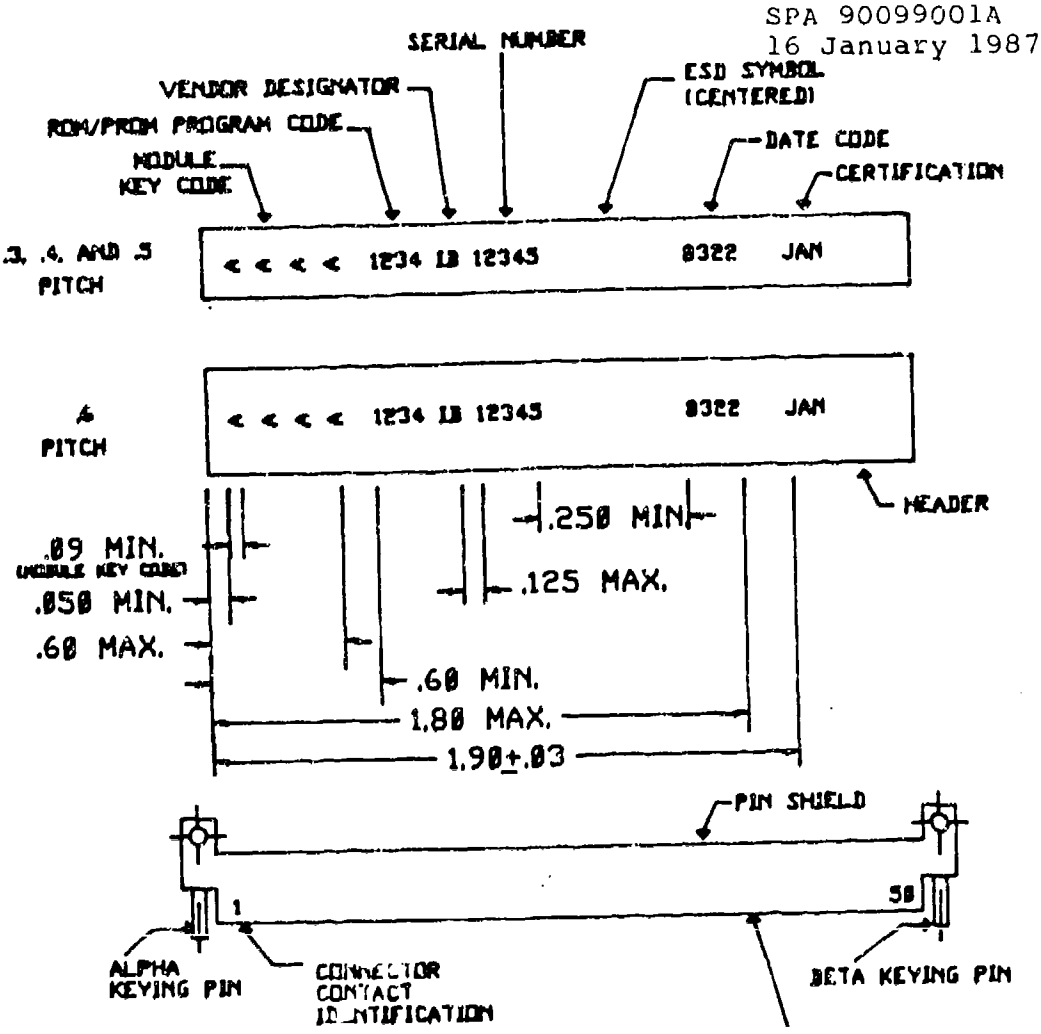
b. Module insertion.

c. Component protection.

d. Test point access.

3.2.1.5.1 Module identification. The module identification header shall have the configuration and marking as specified on figure 2.

3.2.1.5.2 Module insertion. The header structure shall be capable of withstanding 100 pounds (445 newtons) of insertion force.



THESE AREAS RESERVED (EITHER SIDE OF MODULE FOR THE MARKING OF NAME AND PART NUMBER, REVISION LETTER, AMENDMENT NUMBER, RAM/ROM TEST CODE, AND MANUFACTURER'S IDENTIFICATION. "JAN" OR "J" CERTIFICATION MARK SHALL PRECEDE THE PART NUMBER (SEE 3.2.2.3).

INCHES	MM
.03	0.8
.050	1.27
.09	2.3
.125	3.18
.250	6.35
.3	8
.4	10
.5	13
.60	15.2
.6	15
1.80	45.7
1.90	48.3

NOTES:
1. Dimensions are in inches.
2. Metric equivalents are given for information only.

FIGURE 2. Module marking areas.

3.2.1.5.3 Component protection. The header structure shall be designed to help prevent component damage during exposure to insertion and extraction.

3.2.1.5.4 Test point access. The module header shall provide readily accessible test points, however, all final electrical acceptance testing shall be performed through the (input/output) connector pins on the module. All individual modules shall be designed such that the removal of the header shall in no way effect the proper functioning of the module.

3.2.1.6 Pin shields. Modules shall be provided with pin shields to protect the contact pins. Modules shall have pin shields adjacent to each outside row of module pins. Pin shields shall be of a nonconducting material or if of a conducting material, the outside surface of the shield shall be treated in a manner that will prevent conduction into the base conducting material.

3.2.1.6.1 Pin shield retention. The pin shield shall withstand a force of 4 pounds (18 newtons) minimum maintained for 10 to 15 seconds without separation from the module or damage to the pin shield. This requirement shall be met after exposure to all manufacturing process temperatures including preconditioning.

3.2.1.7 Module connector. All connectors shall be in accordance with the requirements specified herein and in MIL-C-28754. The basic module connector shall have two rows of 50 metal bayonet type contact pins. Modules of .4, .5 and .6 inch (10, 13, and 15 mm) pitch may increase contact pin quantity to three, four, and five rows of 50 contact pins with all rows of 50 pins to be complete. Multiple growth modules may increase contact pin row quantities with each row of pins complete. Up to 7 rows of 50 pins is maximum allowable. Modular connectors which support digital, RF, and fiber optic contacts are permitted.

3.2.1.7.1 Connector location. The location of connectors shall be as shown on figure 1. Each connector shall have contacts identified by numbers indicating the first and last pin of the row closest to the pin shield as shown on figure 1.

3.2.1.7.2 Module contact pins. The number of contact pins on the module shall be specified in the detail specification. The contact configuration is controlled only on that part of the contact pin protruding from the module connector surface (the interface plane).

3.2.1.7.2.1 Contact pin location. The location of contact pins shall be as shown on figure 1.

3.2.1.7.2.2 Connector contact integrity. Each contact pin, as mounted in the connector, shall withstand an axial force of 20 ounces (5.6 newtons) minimum applied in 2 to 10 seconds along the length of the contact blade in either direction and maintained for 10 to 15 seconds.

3.2.1.8 Module keying. Each module is assigned an alpha or alpha numeric key code. The first letter indicates the style and angular position of a keying pin in the alpha keying pin location and the last letter designates the style and angular position of a keying pin in the beta location. The following paragraphs provide a sample keying scheme.

3.2.1.8.1 Keying pin locations. There are two keying pin locations on each module designated alpha and beta. The alpha and beta keying pin locations are near the lowest and highest numbered connector pins in the first row, respectively, as shown on figure 3. Keying pins on a mulitple growth module should be located at the extreme ends of the first module increment having a connector.

3.2.1.8.2 Keying pin orientation. Keying pins shall be oriented to agree with the basic angle specified for the module by the code letters on figures 4, 5, or 6. Figure 4 illustrates the module axis and specifies the tolerance for angular positioning.

3.2.1.8.3 Keying pin styles. Keying pin styles shall be in accordance with figure 7.

3.2.1.8.4 Keying pin sets. Only the keying pin styles and keying pin locations in table III are permitted.

3.2.1.8.5 Keying pin integrity requirement. When installed in the module, the keying pins shall meet the following integrity requirements.

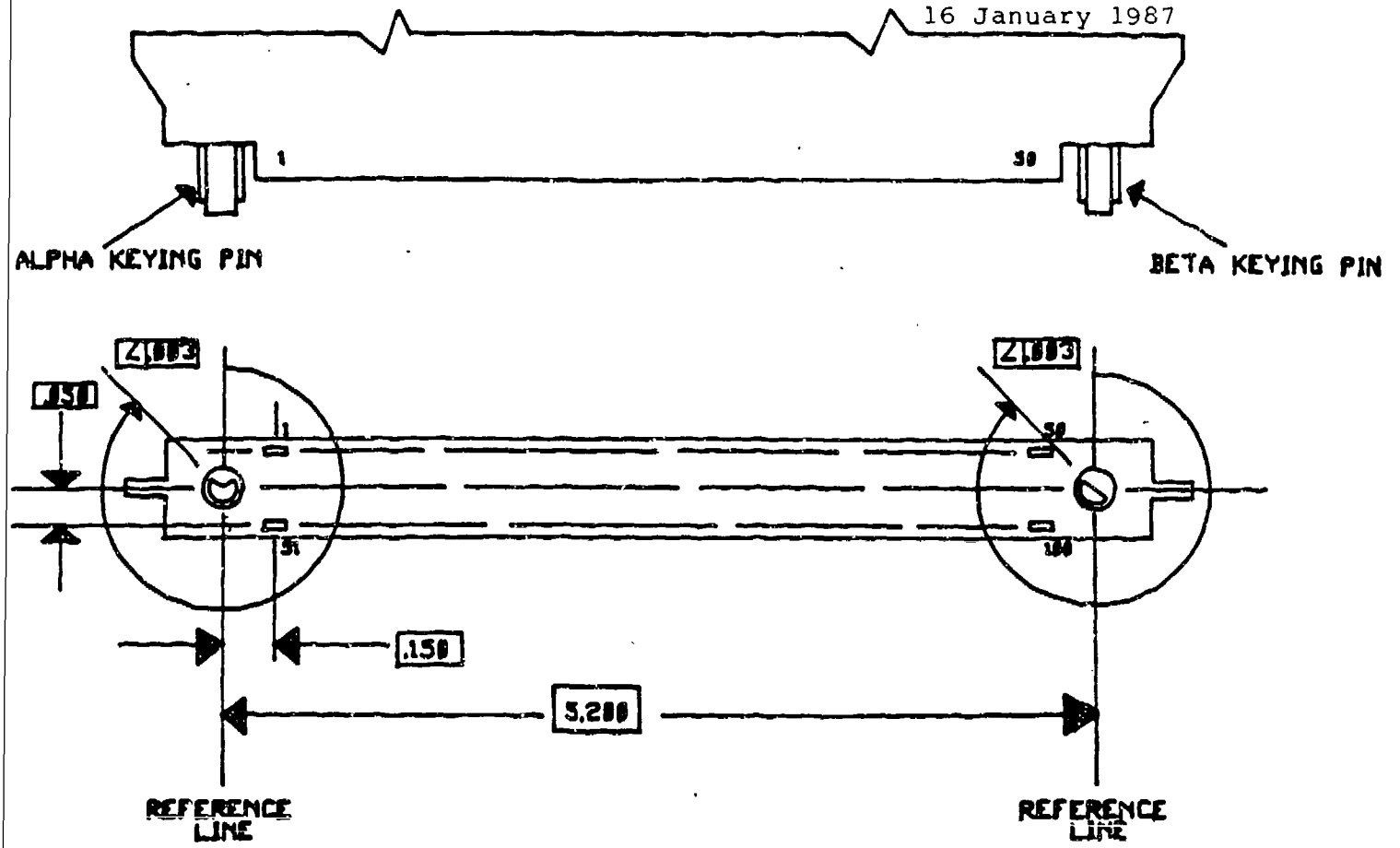
3.2.1.8.5.1 Keying pin torque. Each key pin shall withstand a torque of 20 inch-ounces (0.14 newton-meter) minimum applied in 2 to 10 seconds and maintained for 10 to 15 seconds.

3.2.1.8.5.2 Keying pin pullout. Each key pin shall withstand a pullout force of 9 pounds (40 newton8) minimum applied in 2 to 10 seconds and maintained for 10 to 15 seconds.

3.2.1.8.5.3 Keying pin pushout. Each keying pin shall withstand a pushout force of 40 pounds (178 newtons) applied in 2 to 10 seconds and maintained for 10 to 15 seconds. The force shall be applied in the oppositie direction as the force in 3.2.1.8.5.2.

3.2.1.8.5.4 Keying pin cantilever load. Each key pin shall withstand a cantilever load of 10 pounds (44 newtons) minimum applied in 2 to 10 seconds and maintained for 10 to 15 seconds.

3.2.1.9 Printed wiring boards and printed wiring assemblies. Printed wiring boards and printed-wiring assemblies shall conform to the following requirements.



INCHES	MM
.003	0.08
.050	1.27
.150	3.81
5.200	132.08

NOTES:

1. Dimensions are in inches.
2. Metric equivalents are given for general information only.

FIGURE 3. Keying pin location.

Location		Style of	Notes
Alpha (α)	Beta (β)	Combination	
Style 1	Style 1	Style 1/1	See figure 5
Style 1	Style 2	Style 1/2	See figure 4
Style 2	Style 2	Style 2/2	See figure 6

TABLE III. Keying pin styles and locations.

BETA / LOCATION / ALPHA	A, J 0°	B, K 45°	C, L 90°	D, M 135°	E, N 180°	F, P 225°	G, Q 270°	H, R 315°
A, J 0°	A-A J-A	A-J J-J	A-B J-B	A-K J-K	A-C J-C	A-L J-L	A-D J-D	A-M J-M	A-E J-E	A-N J-N	A-F J-F	A-P J-P	A-G J-G	A-Q J-Q	A-H J-H	A-R J-R
B, K 45°	B-A K-A	B-J K-J	B-B K-B	B-K K-K	B-C K-C	B-L K-L	B-D K-D	B-M K-M	B-E K-E	B-N K-N	B-F K-F	B-P K-P	B-G K-G	B-Q K-Q	B-H K-H	B-R K-R
C, L 90°	C-A L-A	C-J L-J	C-B L-B	C-K L-K	C-C L-C	C-L L-L	C-D L-D	C-M L-M	C-E L-E	C-N L-N	C-F L-F	C-P L-P	C-G L-G	C-Q L-Q	C-H L-H	C-R L-R
D, M 135°	D-A M-A	D-J M-J	D-B M-B	D-K M-K	D-C M-C	D-L M-L	D-D M-D	D-M M-M	D-E M-E	D-N M-N	D-F M-F	D-P M-P	D-G M-G	D-Q M-Q	D-H M-H	D-R M-R
E, N 180°	E-A N-A	E-J N-J	E-B N-B	E-K N-K	E-C N-C	E-L N-L	E-D N-D	E-M N-M	E-E N-E	E-N N-N	E-F N-F	E-P N-P	E-G N-G	E-Q N-Q	E-H N-H	E-R N-R
F, P 225°	F-A P-A	F-J P-J	F-B P-B	F-K P-K	F-C P-C	F-L P-L	F-D P-D	F-M P-M	F-E P-E	F-N P-N	F-F P-F	F-P P-P	F-G P-G	F-Q P-Q	F-H P-H	F-R P-R
G, Q 270°	G-A Q-A	G-J Q-J	G-B Q-B	G-K Q-K	G-C Q-C	G-L Q-L	G-D Q-D	G-M Q-M	G-E Q-E	G-N Q-N	G-F Q-F	G-P Q-P	G-G Q-G	G-Q Q-Q	G-H Q-H	G-R Q-R
H, R 315°	H-A R-A	H-J R-J	H-B R-B	H-K R-K	H-C R-C	H-L R-L	H-D R-D	H-M R-M	H-E R-E	H-N R-N	H-F R-F	H-P R-P	H-G R-G	H-Q R-Q	H-H R-H	H-R R-R

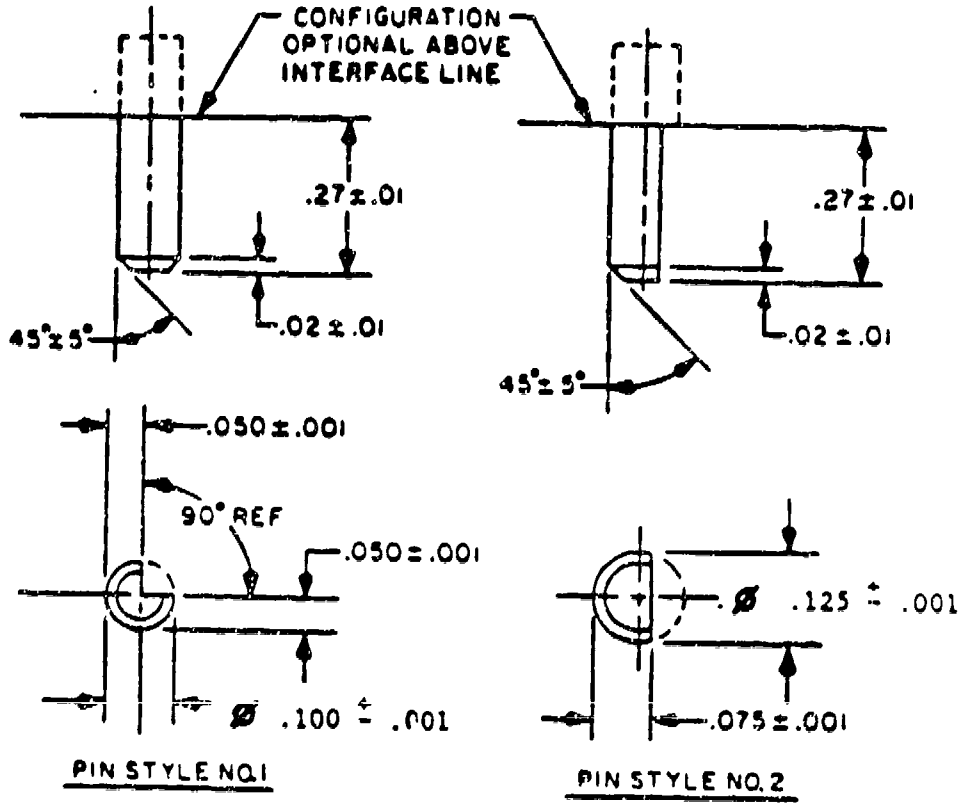
FIGURE 4 Style 1/2 keying chart (viewing connector as shown on figure 3).

KEYS LOCATION ALPHA	S	T	U	V	W	X	Y	Z
0° S								
45° T	T-S	T-T	T-U	T-V	T-W		T-Y	T-Z
90° U	U-S	U-T	U-U	U-V	U-W			U-Z
135° V	V-S	V-T	V-U	V-V	V-W			
180° W		W-T	W-U	W-V	W-W			
225° X			X-U	X-V	X-W			
270° Y				Y-V	Y-W			
Z					Z-W			

FIGURE 5. Style 1/1 keying chart (viewing connector as shown on figure 3).

BETA LOCATION / ALPHA	S 0°	T 45°	U 90°	V 135°	W 180°	X 225°	Y 270°	Z 315°
0° S	S-S	S-T	S-U	S-V		S-X	S-Y	S-Z
45° T								
90° U						U-X		
135° V						V-X	V-Y	
180° W						W-X	W-Y	W-Z
225° X	X-S					X-X	X-Y	X-Z
270° Y	Y-S	Y-T				Y-X	Y-Y	Y-Z
315° Z	Z-S	Z-T	Z-U			Z-X	Z-Y	Z-Z

FIGURE 6. Style 2/2 keying chart (viewing connector as shown on figure 3).



INCHES	MM
.001	0.03
.01	0.3
.02	0.5
.050	1.27
.075	1.90
.100	2.54
.125	3.18
.27	6.9

NOTES:
1. Dimensions are in inches.
2. Metric equivalents are given for general information only.

FIGURE 7. Module keying pin styles.

3.2.1.9.9 Rigid printed-wiring boards. The design of rigid printed-wiring boards shall conform to the requirements specified in MIL-STD-275. Equivalent materials, processes, requirements, and so forth, shall be utilized only when approved by the SEMP-QAA. These equivalent materials, processess, requirements, and so forth, shall be documented and forwarded to the SEMP-QAA for review and approval.

3.2.1.9.2 Printed-wiring assemblies. The design of printed-wiring assemblies shall conform to the requirements specified in MIL-STD-1389, appendix F. Equivalent materials, processes, requirements, and so forth, shall be utilized only when approved by the SEMP-QAA. These equivalent materials, processes, requirements, and so forth, shall be documented and forwarded to the SEMP-QAA for review and approval.

3.2.1.9.3 Flexible printed wiring and assemblies. Flexible printed wiring and assemblies shall conform to the requirements of MIL-P-50884, type B. Performance, quality assurance, and workmanship shall be in accordance with MIL-P-50884.

3.2.1.10 Materials. Materials used in the manufacture of modules shall conform to the requirements specified herein and shall be certified in accordance with applicable specifications where required. When a material is not specified, a material shall be used which will enable the module to satisfy the requirements specified herein. Acceptance or approval of a constituent material shall not be construed as an assurance of the acceptance of the finished product.

3.2.1.10.1 Use of toxic material. Materials which are capable of producing dangerous gasses or other harmful toxic effects as defined in BUMED INSTRUCTION 6270.3 over the temperature range of -55 C to + 125 C, while burining, shall not be used unless suitable nontoxic material is not available.

3.2.1.10.2 Use of flammable material. Materials used in modules shall be in accordance with the requirements of MIL-STD-454, requirement 3, and shall be self extinguishing within 5 seconds after removal of flame.

3.2.1.10.3 Use of material affected by fungus. Materials used in modules shall not support fungal growth when tested in accordance with MIL-STD-810, method 508.

3.2.1.11 Finishes and protective treatments. The finishes and protective treatment of surfaces shall enable the module to meet the requirements specified herein. Acceptance or apporval of a finish or protective treatment shall not be construed as an assurance of the acceptance of the finished product. MIL-E-16400 shall be used in the selection of finishes and protective treatments.

3.2.1.11.1 Module surface finish. The surface finish of the module shall be free of any imperfections that have a detrimental effect upon the performance of the module. The surface of the ribs shall be machined smooth and 25 microinch (0.00064 mm) or better.

3.2.1.11.2 Copper and copper composite frame plating. All copper and copper composite frames shall be electroless nickel plated in accordance with MIL-C-26074, class 1, grade A.

3.2.1.11.3 Aluminum frame plating. All aluminum frames shall receive an anodic treatment in accordance with MIL-A-8625, type III, class 2, black.

3.2.1.11.4 Connector body plating. Any aluminum parts utilized on the connector shall receive an anodic treatment in accordance with MIL-A-8625, type III, class 2, black.

3.2.1.11.5 Conformal coating. The conformal coating shall be in accordance with the requirements of MIL-STD-1389, appendix F. The conformal coating shall be a continuous, homogeneous, fully cured material which covers all components, leads, and circuitry, except grounding surfaces. The coating thickness may vary with the irregularity of the module surface.

3.2.1.12 Weight. Modules shall be designed for minimum weight consistent with reliable circuit operation.

3.2.2 Module marking. All modules shall be identified and marked with appropriate identifiers as specified herein. Figure 2 specifies the marking areas for the following information:

a. Key code.

b. Module part number, revision letter and amendment number.

c. Certification mark.

d. Module name.

e. Serial number.

f. Manufacturer's information.

   1. Manufacturer's identification.

   2. Date code.

g. Connector contact identification.

h. Electrostatic discharge marking (ESD).

i. RAM/ROM test code.

j. ROM/PROM program code.

All markings excluding key code shall be a minimum of 0.06 inch (1.5 mm) high and shall be located as shown on figure 2 and applied in accordance with MIL-STD-130. All marking shall be in a contrasting color to the surrounding module area. All marking shall be permanent and legible in accordance with MIL-STD-1285.

3.2.2.1 Module key code. The key code assigned to each module shall be marked as shown on figure 2. The marking is located at the alpha end of the module on the top surface of the identification header.

3.2.2.2 Module part number and revision status. The module part number, revision letter, and amendment number shall be marked as shown on figure 2. This information is located in the same area as the manufacturer's information. All standard module part numbers will be assigned by the SEMP-DRA. Requests for part number assignment shall be prepared and submitted in accordance with MIL-STD-1378. Modules documented with MIL-STD-961 military specifications shall be marked with the revision status of the specification to which the modules were tested. This marking shall be as follows:

M	28787	/123	-1	A	(1)
Military specification	Specification number	Detail specification number	Environmental class	Detail specification revision number	Detail specification amendment number

For example, a module could be marked M28787/123-1 A(1). Environmental class I shall be indicated by a -1, environmental class II shall be indicated by a -2, environmental class III shall be indicated by a -3, and environmental class IV shall be indicated by a -4. The detail specification revision letter and amendment number are not a part of the module part number and shall be left blank if none exists. The example part number is not intended to designate a length of field requirement. The length of the part number will vary according to the applicable detail specification.

3.2.2.2.1 Module part number and revision status for special modules. Part numbers for special modules, as defined in MIL-STD-1378, will be assigned as directed by the acquisition activity. Special modules shall be marked with the revision status of the document to which the modules were built.

3.2.2.3 Certification mark. All modules that meet the requirements of MIL-M-28787 and the detail specifications shall have the Government certification mark "JAN" or "J" marked as shown on figure 2.

3.2.2.4 Module item name. Each module shall have its item name marked in the area shown on figure 2. The item name and manufacturer's information shall be oriented such that both are readable from the same point of view. The name marked on the module shall agree with the item name in the title of the detail specification, however, abbreviation in accordance with MIL-STD-12 is permissible. The module design activity is responsible for generation of an approved item name. The item names in H6 shall be used if they appropriately describe the module. When H6 does not list an item name which appropriately describes the module, an item name shall be developed in accordance with MIL-STD-961. The SEMP-DRA is the approval source for item names.

3.2.2.5 Manufacturer's information. The following information shall be marked on each module at the locations shown on figure 2. No other module manufacturer's part number shall be marked on the module.

3.2.2.5.1 Manufacturer's identification. Each module shall be marked with either the manufacturer's identification code or manufacturer's name. The manufacturer's code, if utilized, shall be a numerical code as listed in H4-2. A test code assigned by the SEMP-QAA for each vendor's integrated circuit type used shall be marked on each RAM/ROM module in the area specified for the manufacturer's identification.

3.2.2.5.2 Serial number. Each module shall have a serial number including vendor's designation. The serial number is located on the top surface of the same fin/header used for marking the key code. The serial number shall consist of five digits with significant digits prefixed with zeros as required. The serial number shall be affixed to the module prior to electrical acceptance test.

3.2.2.5.3 Serial number sequence. Each module manufacturer shall serialize each module manufactured under the requirements of the SEMP. The serial number for any given key code will start with number 1 and continue in numerical sequence as many times as the module is manufactured regardless of contract or customer.

3.2.2.5.4 Vendor designation. A single or double alpha character shall be assigned to each module manufacturer contracted to produce modules. The designation shall be prefixed to the module serial number. Request for a vendor designation shall be submitted to the SEMP-DRA.

3.2.2.6 Date code. Each module shall be marked with a four digit date code designating the year and the week of manufacture. The first two digits of the code shall be the last two numbers of the year and the third and fourth digits of the code shall be the calendar week. When the number of the week is a single digit, it shall be preceded by a zero. The date code for a given module shall be the calendar week in which the last major manufacturing assembly process occurred prior to the final acceptance inspection plus or minus one week.

3.2.2.7 Electrostatic discharge (ESD). Modules that are determined by the SEMP-QAA to require special handling due to sensitivity to electrostatic discharge (ESD) by prior knowledge of device technologies or by testing to MIL-M-28787 shall be marked in the areas shown on figure 2 with the sensitive electronic device symbol specified in MIL-STD-1285. If the minimum symbol size specified in MIL-STD-129 can not be met, the size shall be maximized for the particular fin configuration.

3.2.3 Module mechanical requirements. The following mechanical requirements apply.

3.2.3.1 Module integrity. Each module, with the connector assembled shall withstand without damage or separation on minimum axial force normal to the interface plane equal to 100 pounds (445 newtons) on insertion and 4 ounces (1.11 newton) per contact on extraction. The total computed force shall be applied to the module to simulate module insertion and extraction. The force shall be applied in 2 to 10 seconds and maintained for 10 to 15 seconds.

3.2.3.1.1 Module extractor integrity. The extractor structure shall provide the strength required to extract the module and meet the requirements of 3.2.3.1.

3.2.3.1.2 Module header integrity. The header structure shall provide the strength required to install the module and meet the requirements of 3.2.1.5.2.

3.2.3.2 Module torque. The module shall be capable of withstanding a 6 inch-pound (0.68 newton-meter) torque applied in 2 to 10 seconds and maintained for 10 to 15 seconds in both directions along the header in a direction perpendicular to the plane of the header without detrimental effect to the mechanical or electrical properties of the module. During the time the torque is applied, the module shall be rigidly supported within a zone between the interface plane and 0.5 inch (13 mm) above the interface plane.

3.2.3.3 Module cantilever load. The module shall be capable of withstanding a force of 2 pounds (9 newtons) applied perpendicular to the header height along the centerline midway between the two extractor holes. The force shall be applied in two directions and shall be applied in 2 to 10 seconds and maintained for 10 to 15 seconds without detrimental effect to the header structure.

3.2.3.4 Durability. The module shall be capable of withstanding 500 cycles of mating and unmating with no degradation of module performance. The module shall also be capable of withstanding 500 cycles of lateral displacement to simulate the use of thermal clamping devices. The lateral displacements may be included in the insertion/extraction sequences or completed after the insertion/extraction cycling.

3.2.4 Module electronic design requirements. Modules shall be designed in accordance with the following requirements.

3.2.4.1 Personnel safety. Modules with voltages exceeding 30 volts (direct current or alternating current root mean square) to ground shall have exposed conductive frame surfaces (except for pin shields and key pins) connected to either the 0 volt or chassis ground contact pin. The maximum resistance between the ground pin and the exposed conductive frame surfaces shall be 1 ohm.

3.2.4.2 Powered socket. The detail specification for modules containing device technologies which cannot be protected by the module design during removal or insertion into a powered socket must contain caution notices of susceptability to damage.

3.2.4.3 Component selection. Electronic components and hardware used in modules shall have a demonstrated quality level and environmental performance equivalent to or better than that of available military parts. Nonhermetically sealed packaged relays and semiconductor devices having hermetically sealed equivalents shall not be used.

3.2.4.3.1 Germanium semiconductors. Germanium semiconductors shall not be used.

3.2.4.3.2 Discrete semiconductors. Discrete semiconductors shall be in accordance with the requirements of MIL-S-19500 and shall be selected according to the following priority list. Devices listed in b, c, and d shall be approved by the SEMP-QAA prior to use.

a. MIL-S-19500 JANTX devices listed in MIL-STD-242.

b. Other MIL-S-19500 JANTX devices.

c. Devices being considered for a MIL-S-19500 JANTX detail specification. Devices shall be equal to or better than MIL-S-19500 devices.

d. Other devices. Devices shall be equal to or better than MIL-S-19500 devices.

3.2.4.3.3 Integrated circuits. Integrated circuits shall be in accordance with the following requirements.

3.2.4.3.3.1 Quality requirements. Integrated circuits shall be in accordance with the requirements of MIL-M-38510, class B. The module supplier shall use MIL-M-38510 JAN QPL devices when available or procure other devices with equivalent specifications. All equivalent specifications shall be submitted to the SEMP-QAA for approval prior to initial qualification. Equivalent specifications shall include:

a. The screening shall be in accordance with MIL-STD-883, method 5004, class B.

b. Quality conformance shall be demonstrated in accordance with MIL-STD-883, method 5005, groups A, B, C, D, and E (if applicable), class B.

c. Generic data is acceptable for demonstrating quality conformance in accordance with MIL-STD-883, method 5005, groups C and D, class B. A generic family shall be electrically and structurally similar integrated circuits. They are designed to perform the same type of basic circuit function using the same basic circuit element configuration and differ only in the number of identically specified circuits which they contain. They are designed for the same supply, bias, and signal voltages and for input-output compatibility with each other under an established set of loading rules. They are enclosed in packages of the same construction and outline, differing only in the number of active external package leads included or used and made from the same materials by use of the same processes.

3.2.4.3.3.2 Selection requirements. Integrated circuits shall be in accordance with the requirements of MIL-M-38510 and shall be selected according to the following priority list. Devices listed in b, c, and d shall be approved by the SEMP-QAA prior to use.

a. MIL-M-38510 JAN microcircuits listed in MIL-STD-242.

b. Other MIL-M-38510 JAN microcircuits.

c. DESC Selected Item Drawing Microcircuits.

d. Other microcircuit devices shall be equal to or better than MIL-M-38510 JAN devices.

3.2.4.3.3.3 Restricted usage. Memory devices of identical size and configuration from different suppliers shall not be mixed on individual modules.

3.2.4.3.3.4 Substitution requirements. A MIL-M-38510 part may be substituted if the quality requirements are met for a vendor approved Specification Control Drawing (SCD) or DESC drawing procured part. A DESC drawing procured part may be substituted if the quality requirements are met for the module vendor approved SCD.

3.2.4.3.4 Passive components. Passive components shall be selected according to the following priority list. Devices listed in c and d shall be approved by the SEMP-QAA prior to use.

a. Established reliability (ER) specification parts (minimum level P if multiple sources exist) listed in MIL-STD-242.

b. ER parts (minimum level M if required to achieve multiple sources) listed in MIL-STD-242.

c. Other ER parts.

d. Conventional military specification parts.

e. ER parts that use the Weibull failure rate prediction method, such as MIL-C-39003, shall require a "B" minimum failure rate level.

3.2.4.3.5 Hybrid microcircuits. Hybrid microcircuits shall be in accordance with the requirements of MIL-STD-883 and MIL-M-38510, or equivalent.

a. Hybrid microcircuits which are contained in packages having an inner seal perimeter of 2.0 inches (51 mm) or greater shall be in accordance with the requirements of MIL-STD-883, method 5008.

b. Hybrid microcircuits which are contained in packages having a seal perimeter of less than 2.0 inches (51 mm) shall be in accordance with the requirements of MIL-STD-883, method 5004 and 5005, class B, or method 5008.

All equivalent specifications shall be submitted to the SEMP-QAA for approval prior to initial qualification.

3.2.5 Thermal requirements. The following thermal requirements apply.

3.2.5.1 Heat dissipation. Modules shall be designed to ensure that critical component temperatures are not exceeded when modules are operated at typical power at the maximum thermal interface temperature for the appropriate class.

3.2.5.2 Typical power dissipation. Typical power dissipation means the maximum recommended power dissipation under nominal module operating condition. Typical power values for semiconductor devices are derived from contractor developed characterization data (if available) or secondly, from vendor data sheets. When the typical power dissipation for a component cannot be determined, the maximum power dissipation for worst case module operating conditions shall be used.

3.2.5.3 Component temperatures. The following requirements for critical component temperature (CCT) and transient critical component temperature (TCCT) apply.

3.2.5.3.1 CCT. The CCT for semiconductor devices dissipating 2.5 watts or less typical power shall be 70 C junction for classes I and III and 90 C junction for classes II and IV. For semiconductors dissipating more than 2.5 watts typical power, the CCT may increase 15 C/watt or a maximum of 15 C above that specified for less than 2.5 watt devices. For all other components, the CCT shall be equal to the individual components maximum specified operating temperature minus 30 C and shall be specified on the component's hottest external area.

3.2.5.3.2 TCCT. The TCCT for all devices shall be the appropriate CCT plus 20 C.

3.2.6 Environmental requirements. The module shall meet the environmental requirements of MIL-M-28787, for class II or class IV modules except as modified herein.

3.2.6.1 Life test. When tested in accordance with paragraph 4.1.1 the end of life requirements shall be in accordance with the detailed module specification.

3.2.6.2 Inclination. When specified in the detailed module specification, modules shall be capable of proper operation during the test requirements of MIL-M-28787. Modules employing heat pipes shall meet the requirements for operating temperature after being subjected to the test requirements of paragraph 4.1.2.

3.2.6.3 Thermal shock. Modules shall be capable of proper operation and shall show no deterioration after being subjected to the test requirements of paragraph 4.1.3.

3.2.6.4 Salt fog. Modules shall be capable of proper operation and shall show no deterioration after being subjected to the test requirements of paragraph 4.1.4.

3.2.7 Documentation requirements. Individual module designs shall be fully documented in accordance with this specification. Module specifications prepared in accordance with this document shall be the governing documents used for the procurement and testing of modules. Information, in addition to that required by this specification, may be added to the detailed module specification as deemed necessary for procurement and testing of a particular module.

3.2.7.1 Specification classification. Individual module specifications shall be prepared in accordance with MIL-STD-961.

3.2.7.2 Types of specifications. Module specifications shall be either Type C2a Format or Type C2b Format in accordance with MIL-S-83490 for standard or special modules respectively. All specifications shall be prepared as book form drawings on "A" size drawing forms in accordance with DOD-STD-100.

3.2.8 Preconditioning. All modules shall be subjected to preconditioning in accordance with the following:

a. The module shall be subjected to nonoperating temperature cycling for a minimum of ten complete cycles of temperature variation. A cycle shall consist of 15 minutes at temperature extremes of +85 C, or above, and -55 C, or below, with a maximum transfer time between temperature extremes of 5 minutes. A cycle may begin at either temperature.

b. Upon completion of the temperature cycling, the module shall meet the initial electrical requirements specified in the detailed module specification. Preconditioning shall be completed prior to the 25 C electrical inspection.

3.2.9 Life expectancy. Modules shall be designed for a minimum life expectancy of 100,000 hours operation at maximum temperature for the appropriate class.

3.2.10 Workmanship. Workmanship shall be of such quality that the module will comply with the requirements specified in the detailed module specification and MIL-STD-454, requirement 9.

4.0 QUALITY ASSURANCE PROVISIONS.

4.1 Exceptions. The quality assurance provisions shall be in accordance with MIL-M-28787 except for the following:

4.1.1 Life. The module shall be mounted in a suitable test fixture and operated for a period of not less than 500 hours at 85 C thermal interface temperature as specified in the detail module specification. Upon completing the life test, and while still at 85 C, the module shall meet the end-of-life 85 C electrical requirements defined in the individual module specification. The module shall then be returned nonoperating to a thermal interface temperature of 25 C at a chamber temperature rate not to exceed 1 C per minute. After a stabilization period of four hours, the module shall be tested to and meet the end-of-life requirements specified in the detailed module specification.

4.1.2 Heat pipes. Modules employing heat pipes for cooling shall meet operating temperature requirements when the module heat sink is inclined at an angle of 90 degrees from the horizontal.

4.1.3 Thermal shock. The module shall be tested in accordance with MIL-STD-202, test method 107, for 400 cycles, -55 C to +125 C. The module shall pass all electrical tests. There shall be no evidence of deterioration or physical damage after thermal shock.

4.1.4 Salt fog. The module shall be tested in accordance with MIL-STD-810, method 509, procedure I. The module shall be examined with the aid of a 10-power magnification following a gentle wash in warm (37 C $\pm$ 5 C) water upon removal from test chamber (to remove visible salt deposits), and storage for 48 hours at room ambient conditions to allow for evaporation of excess moisture. Failure mechanisms shall include pits, crack formations, intergranular attack, etc.: that is, any concentrated attack that weakens the cross section. Surface corrosion products shall not be evidence of failure. The module shall be subjected to all electrical tests. Any failures due to corrosion or corrosion products shall be cause for failure.

5.0 PACKAGING.

5.1 General. Packing and packaging of modules shall be in accordance with MIL-M-28787 (see paragraph 5.0).

6.0 NOTES.

6.1 Intended use. Modules specified herein are intended for use in military systems and subsystems.

6.2 Definitions.

6.2.1 Standard Electronic Modules Program (SEMP). A design standardization program which has for its goal the development of functional electronic modules from which a variety of complex military electronic systems may be built.

6.2.2 SEMP Design Review Activity (SEMP-DRA). The activity responsible for the review and classification of module designs is the Naval Avionics Center, 6000 E. 21st Street, Indianapolis, Indiana 46218 (Code 965).

6.2.3 SEMP Quality Assurance Activity (SEMP-QAA). The activity responsible for specification and design review, correlation, vendor audits, and qualification testing is the Naval Weapons Support Center, Crane, Indiana 47522 (Code 603).

6.2.4 Key code. An alpha or alpha numeric (e.g. A4A) designator used to identify the style and angular position of the keying pins.

6.2.5 Alpha end. The end of the module nearest the lowest numbered pin.

6.2.6 Beta end. The end of the module farthest from the lowest numbered pin.

6.2.7 Critical component temperature (CCT). The maximum temperature allowed for any component in the module while the module is operating at maximum class temperature.

6.2.8 Transient critical component temperature (TCCT). The maximum temperature allowed for any component in the module while the module is operated at maximum class temperature plus 20 C without exceeding any individual component TCCT.

6.2.9 End-of-life tolerance. The minimum and maximum limits for any particular module characteristic after being subjected to 100,000 hours operation established at an ambient temperature of 25 C $\pm$ 5 C as well as over the entire temperature range specified for the module.

6.2.10 Powered socket. A socket whose terminals are connected to active power supplies, control circuits, loads, and signal sources to simulate system requirements.

VHSIC Phase 2 INTEROPERABILITY STANDARDS

PI-Bus SPECIFICATION

September 8, 1986

Version 2.0

IBM

Honeywell

TRW

Copyright 1985, 1986

IBM Honeywell TRW

Copying in printed form is permitted without payment of royalty provided that (1) each reproduction is done without alteration, (2) each reproduction is complete and (3) the copyright notice is included.

CONTENTS

Section	Title	Page
1.0	SCOPE	1-1
1.1	SCOPE.	1-1
1.2	PURPOSE.	1-1
1.3	INTENDED APPLICATION.	1-1
1.4	CLASSIFICATION.	1-1
2.0	APPLICABLE DOCUMENTS	2-1
2.1	GOVERNMENT DOCUMENTS.	2-1
2.2	NON-GOVERNMENT DOCUMENTS.	2-1
3.0	DEFINITIONS	3-1
3.1	ITEM DEFINITION.	3-1
3.2	TERM DEFINITIONS.	3-3
4.0	PHYSICAL LAYER	4-1
4.1	INTRODUCTION.	4-1
4.2	LINE DEFINITION.	4-1
4.2.1	Nomenclature.	4-1
4.2.2	Bused Signal Lines.	4-1
4.2.3	Bus Clock.	4-8
4.2.4	Module Identification.	4-8
4.3	ELECTRICAL REQUIREMENTS	4-9
4.3.1	Backplane Requirements.	4-9
4.3.2	Module Requirements.	4-10
5.0	DATA LINK LAYER	5-1
5.1	INTRODUCTION.	5-1
5.2	GENERAL REQUIREMENTS.	5-1
5.2.1	Introduction.	5-1
5.2.2	Protocol State Transitions.	5-4
5.2.3	Generic Message.	5-7
5.3	DETAILED REQUIREMENTS.	5-25
5.3.1	Introduction	5-25
5.3.2	Bus State Definitions.	5-25
5.3.3	Bus Mastership.	5-29
5.3.4	Message Sequences	5-38
5.3.5	Exception Sequences.	5-67
5.3.6	Wait.	5-75
5.3.7	Data Link Facilities.	5-77
5.3.8	Initialization.	5-87
5.3.9	Error Detection, Recovery And Diagnostics.	5-88

CONTENTS (continued)

LIST OF ILLUSTRATIONS

Figure	Title	Page
Figure 3-1.	Conceptual Model Of Bus And Modules	3-2
Figure 4-1.	Set-up And Hold Timing	4-12
Figure 4-2.	Signal Output Test Circuit	4-13
Figure 4-3.	Output Signal Timing	4-13
Figure 5-1.	PI-bus Protocol State Diagram	5-7
Figure 5-2.	Header Word A Format - Data Lines	5-13
Figure 5-3.	Single Slave Acknowledge Word Format - Data Lines	5-18
Figure 5-4.	Multiple Slave Status Word Format (AWM0) - Data Lines	5-21
Figure 5-5.	Multiple Slave Status Word Format - Data Check Lines	5-22
Figure 5-6.	Tenure Pass Message Header Word Formats	5-34
Figure 5-7.	Parameter Write Header Word Formats	5-39
Figure 5-8.	Block Message - Short Header Word Formats	5-45
Figure 5-9.	Block Message - Extended Header Word Formats	5-53
Figure 5-10.	Bus Interface Message Header Word Formats	5-63
Figure 5-11.	Multicast Acknowledge Register Word Format	5-80
Figure 5-12.	Control Register Word Format	5-81
Figure 5-13.	Module Capabilities Register Word Format	5-82
Figure 5-14.	Vie Interval A Register Word Format	5-83
Figure 5-15.	Vie Interval B Register Word Format	5-84
Figure 5-16.	Vie Priority Register Word Format	5-85
Figure 5-17.	Logical Slave Identification Register Word Format	5-87

CONTENTS (continued)

LIST OF TABLES

Table	Title	Page
Table 4-1.	PI-bus Signal Line Requirements By Type And Class	4-3
Table 4-2.	PI-bus Cycle Types and Valid Symbols	4-6
Table 4-3.	Acknowledge Line Valid Symbols	4-7
Table 5-1.	PI-bus Communications Sequences	5-2
Table 5-2.	PI-bus Protocol States	5-3
Table 5-3.	Generic PI-bus Message Sequence	5-9
Table 5-4.	Message Type Codes	5-14
Table 5-5.	Access Type Codes	5-15
Table 5-6.	Multiple Slave Acknowledge Symbol Formats (Four Words)	5-23
Table 5-7.	Multiple Slave Acknowledge Symbols (Four Words)	5-24
Table 5-8.	Bus State Definitions	5-26
Table 5-9.	Vie Sequence	5-30
Table 5-10.	Module Vie Code Format - Data Lines	5-32
Table 5-11.	Module Vie Code Format - Data Check Lines	5-33
Table 5-12.	Tenure Pass Message Sequence	5-36
Table 5-13.	Parameter Write Sequence - Type 16 Single Slave	5-40
Table 5-14.	Parameter Write Sequence - Type 16 Multiple Slave	5-41
Table 5-15.	Parameter Write Sequence - Type 32 Single Slave	5-42
Table 5-16.	Parameter Write Sequence - Type 32 Multiple Slave	5-43
Table 5-17.	Block Message - SH Sequence - Type 16 Single Slave	5-46
Table 5-18.	Block Message - SH Sequence - Type 16 Multiple Slave	5-47
Table 5-19.	Block Message - SH Sequence - Type 32 Single Slave	5-49
Table 5-20.	Block Message - SH Sequence - Type 32 Multiple Slave	5-50
Table 5-21.	Block Message - EH Sequence - Type 16 Single Slave	5-54
Table 5-22.	Block Message - EH Sequence - Type 16 Multiple Slave	5-56
Table 5-23.	Block Message - EH Sequence - Type 32 Single Slave	5-58
Table 5-24.	Block Message - EH Sequence - Type 32 Multiple Slave	5-60
Table 5-25.	Bus Interface Message Sequence - Type 16 Single Slave	5-64
Table 5-26.	Bus Interface Message Sequence - Type 16 Multiple Slave	5-65
Table 5-27.	Suspend - Short Data Sequence - Type 16 Single Slave	5-69
Table 5-28.	Suspend - Short Data Sequence - Type 32 Single Slave	5-70
Table 5-29.	Suspend - Extended Data Sequence - Type 16 Single Slave	5-71
Table 5-30.	Suspend - Extended Data Sequence - Type 32 Single Slave	5-72
Table 5-31.	Suspend - Type 16 Multiple Slave	5-73
Table 5-32.	Suspend - Type 32 Multiple Slave	5-74
Table 5-33.	Abort Sequence	5-75
Table 5-34.	Data Link Register Address Space	5-78
Table 5-35.	Interpretation and Response for Uncorrectable Invalid Symbols	5-89
Table 5-36.	Slave Response to Cycle Type Sequence Deviations	5-91
Table 5-37.	Cycle Type Deviation Response Definitions	5-92
Table 5-38.	Sequence Error Response	5-93
Table 5-39.	Semantic Errors	5-94

ABSTRACT

This specification defines a linear, multi-drop, synchronous bus (PI-bus) which supports digital message communications between up to 32 modules residing on a single backplane. Messages are transferred datum serial and bit parallel using a datum size of 16 bits (single word) or 32 bits (double word).

The PI-bus uses a master-slave communications protocol which allows the bus master to read data from one slave or write data to any number of slaves in a single message sequence. Messages may be routed to particular modules using either logical or physical addressing. A number of independent messages may be transmitted during a bus master's tenure. The message formats provide a 32 bit virtual address range for each module.

The PI-bus protocol specifies a set of bus state transitions which control the communication sequences and allow the bus to operate in a pipelined manner at the maximum clock rate allowed by the bus signal propagation delay. Master-slave handshaking is provided with a minimal performance penalty by operating the slave modules in synchronism with the master and using bus state look-ahead.

A technique for temporarily suspending low priority block data transfers to reduce bus acquisition latency for higher priority messages is defined.

Bus mastership may be changed either by direct assignment or by priority arbitration. The protocol defines 128 logical levels of message priority and 32 levels of physical priority.

Extensive signal line and sequence error detection capability is incorporated into the bus definition. In addition, an optional single line error correction capability is specified.

PREFACE

This document was jointly prepared by IBM, Honeywell and TRW in partial fulfillment of Contract Data Requirements List (CDRL) item A011 for work being performed under VHSIC Phase 2 Submicrometer Technology Development contracts DAAK20-85-C-0367, F33615-84-C-1500 and N00039-85-C-0111, respectively.

Section 1

SCOPE

1.1 SCOPE.

This specification states the physical, electrical, functional and performance requirements defined for the PI-bus.

1.2 PURPOSE.

The purpose of this standard is to establish requirements for the PI-bus and facilitate interoperability of modules which use the PI-bus.

1.3 INTENDED APPLICATION.

The PI-bus is intended to provide a master-slave communications path for transferring digital messages between a set of up to 32 modules residing on a single backplane.

1.4 CLASSIFICATION.

Bus configurations and modules which conform to this standard may be any of the types, classes and features specified below:

Type 16	16 bit data transfers
Type 32	32 bit data transfers
Class ED	Error Detecting
Class EC	Error Correcting
Feature SO	Slave Only operation
Feature MS	Master and Slave operation

Buses and modules shall be classified according to their maximum capabilities. Bus sequences shall be classified according to the Type or Class of transfer actually used.

All modules and buses shall be capable of operating in Type 16, Class ED mode. Type 32 and Type 16 modules shall be interoperable on a Type 32 bus where the Type 32 modules may communicate using 16 or 32 bit transfers but only 16 bit transfers are used whenever a Type 16 module is an active participant. All active modules on a given bus shall operate in the same class.

Section 2

APPLICABLE DOCUMENTS

2.1 GOVERNMENT DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None.

2.2 NON-GOVERNMENT DOCUMENTS.

The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superseding requirement.

- None.

Section 3

DEFINITIONS

The definitions listed herein shall apply to the PI-bus and PI-bus modules.

3.1 ITEM DEFINITION.

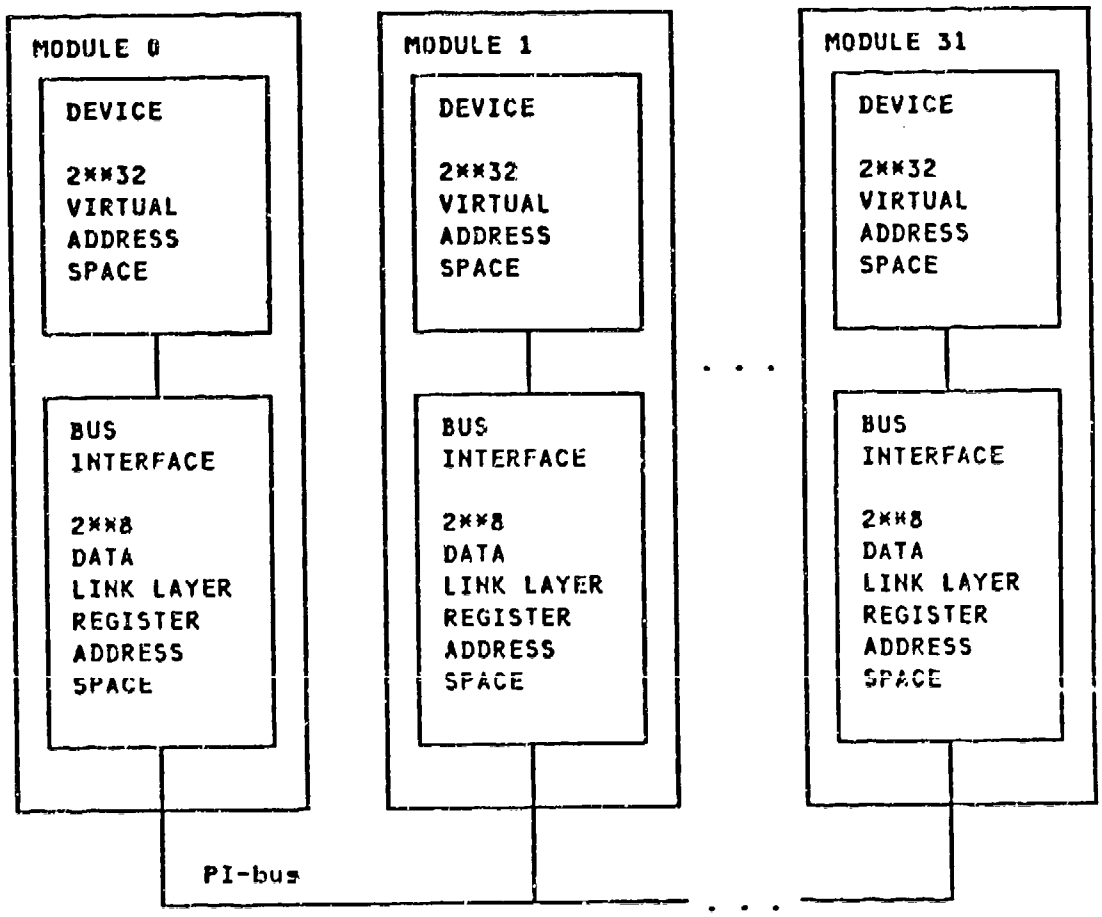
The PI-bus is a linear, multi-drop communications medium which transfers datum serial, bit parallel information among up to 32 modules residing on a single backplane. The datum size may be a single word or a double word.

PI-bus modules are those modules which implement the slave only or master and slave portions of the PI-bus protocol as specified herein.

Figure 3-1, illustrates the PI-bus and PI-bus modules. Conceptually, each module consists of a device which performs the application specific function of the module and a Bus Interface which implements the PI-bus master-slave communications protocol.

The device portion of each module is modelad as a virtual memory space with a 32 bit address range. The Bus Interface is modeled as a separate memory space with an 8 bit Data Link register address range. A separate, 8 bit virtual address called the slave ID is used by the bus master to select one or more modules to participate in a particular communications sequence as slave(s).

Figure 3-1. Conceptual Model Of Bus And Modules



MODULE VIRTUAL ADDRESS SPACE = 2**8
- PHYSICAL SLAVE ID 0 - 31
- LOGICAL SLAVE ID 32 - 255

3.2 TERM DEFINITIONS.

The definitions given below shall apply to the PI-bus and PI-bus modules.

//	The concatenation operator for groups of bits.
active Bus Interface	A Bus Interface that is connected to the bus media, and is currently capable of (and not inhibited from) participating in bus transactions.
arbitration	The process by which a single bus master is selected from competing potential bus masters.
assert (signal)	The action of changing the state of a bus signal line from released, logic 0, to asserted, logic 1, or of ensuring that the line remains in the asserted state.
asserted (signal)	The logic 1 state of a bus signal line. The least positive of the two states of a bus signal line.
backplane	A motherboard comprising wiring for the bus and connectors to the modules attached to the bus.
broadcast	A mode of operation where the bus master transmits data to all modules during a single transfer.
bus acquisition latency	The time from the highest priority module's request for bus mastership to the time at which that module becomes bus master.
bus master	The module currently in control of the bus.
bus tenure	The period between the time a bus master gains control of the bus and the time at which control is released.
contender	A potential bus master module which is actively vying for bus mastership.
device	The portion of a module, excluding the Bus Interface, which does the application depend-

ent function of the module.

double word An ordered set of 32 bits operated on as a pair of words or as a single unit. The most significant bit of a double word is labeled bit 31 and the least significant is labeled bit 0.

linear bus A bus with a single shared medium segment.

message A set of sequences starting with a header and terminating when all bus actions indicated by that header have been performed.

module An entity which is addressable via the bus and has a single connection to the bus.

multicast A mode of operation where the master transmits data to more than one slave during a single transfer.

non-transfer cycle A bus cycle that immediately follows a bus cycle in which a valid Wait is asserted or one of the VZ, HZ, DZ or HAZ cycles specified in the protocol. Information on the Data lines is not used during a non-transfer cycle.

partial message A sequence starting with a header and terminating prematurely due to a suspend, abort or other exception indication prior to a normal completion.

post (symbol) The action taken to assert and/or release individual bus lines within one particular group of bus lines such that either the associated symbol appears on the group or another symbol appears that is the result of individual line ORing of simultaneously posted symbols.

release (signal) The action of ceasing to assert a logic 1 on a bus signal line. The action of releasing a signal line produces a change in the state of the signal line only if no module is asserting that signal.

released (signal) The logic 0 state of a bus signal line produced when no module asserts the signal associated with that line. The more positive of the two states of a bus signal line relative

to the 0 Volt logic reference.

sequence A transaction comprising a number of ordered transfers performing one intended function.

slave A module which is selected by the bus master to participate in a message sequence.

symbol A unit of information on a particular group of bus lines, as represented by a particular binary encoding of bits. A valid symbol is one which conforms to the signal definitions herein including correct parity, Hamming encoding or redundant coding as applicable.

transfer A set of elemental operations on the bus which results in the communication of a bit parallel datum unit between the current bus master and the selected slave(s). The datum unit is either 16 bits (Type 16) or 32 bits (Type 32). See sequence.

word An ordered set of 16 bits operated on as a unit. The most significant bit is labeled bit 15 and the least significant bit is labeled bit 0.

Section 4

PHYSICAL LAYER

4.1 INTRODUCTION.

The physical layer of the PI-bus is specified herein. Signal lines required to implement the PI-bus are defined, including those used for signal line error detection and correction. The electrical characteristics of the module interfaces and backplane are specified and timing definitions are presented.

4.2 LINE DEFINITION.

The PI-bus signal, clock and module identification lines are defined in this section. In addition, the encoding used to achieve signal line error detection and correction is specified by definition of the valid symbols allowed for each signal line group.

4.2.1 NOMENCLATURE.

Lines shall be designated by name or by capital letter abbreviations, e.g. Data or D. Where a set of related lines are represented by the same name, the lines within the set shall be differentiated by number with the least significant bit numbered 0. The nomenclature for single lines shall be the letter abbreviation for the line name followed by the bit number enclosed in < >, e.g. D<0>. The nomenclature X<m..n> shall be the abbreviation for the set of lines Xm to Xn, inclusive, where X is the letter abbreviation for the line name and m and n are the most and least significant bit numbers, respectively. Thus, D<7..0>, represents the least significant eight Data lines. In addition, X<i,j,...,k> shall be an abbreviation for the set of lines X<i>, X<j>,..., X<k>. Thus D<5,3,1,7> stands for the set of lines D<5>, D<3>, D<1>, and D<7>.

P(X) shall designate the parity (modulo 2 sum) of the set of signal lines defined by X. Thus P(D<15..0>,DC<0>) designates the parity of the sixteen bits present on the Data lines plus the Data Check line and P(D<15..0>,DC<0>)=0 represents even parity over the specified lines.

4.2.2 BUSED SIGNAL LINES.

All PI-bus signal lines shall be implemented as wired-or lines bused between modules on a common backplane. Modules and buses shall implement those bused signal lines specified for their particular Type and Class by Table 4-1. Any implemented bus signal lines which are not required during an operation of a particular Type and Class shall be released during that operation.

Symbols posted onto signal lines shall be valid symbols as specified in this section, except that during diagnostic bus operation some invalid symbols are allowed as specified in "5.3.9.5 Diagnostics.."

As a required device function, PI-bus modules shall provide a command path independent of the PI-bus which provides a way to force all PI-bus signal lines to be released by the module. This capability may be used in bus diagnostics and fault isolation. In addition, no signal line shall be asserted from the time power is applied to the PI-bus module until the module has completed reset as defined in "5.3.8 Initialization.." Modules shall not assert any signal line in violation of this specification during power failure.

Table 4-1. PI-bus Signal Line Requirements By Type And Class

	Lines Required (Yes/No)			
	Type 16		Type 32	
NAME	Class ED	Class EC	Class ED	Class EC
Data (D)				
D<31..16>	No	No	Yes	Yes
D<15..0>	Yes	Yes	Yes	Yes
Data Check (DC)				
DC<7..2>	No	Yes	No	Yes
DC<1>	No	Yes	Yes	Yes
DC<0>	Yes	Yes	Yes	Yes
Cycle Type (CT)				
CT<2..0>	Yes	Yes	Yes	Yes
CT Check (CTC)				
CTC<2,1>	No	Yes	No	Yes
CTC<0>	Yes	Yes	Yes	Yes
Acknowledge Set (AS)				
AS<5,4>	No	Yes	No	Yes
AS<3..0>	Yes	Yes	Yes	Yes
Wait (W)				
W<2>	No	Yes	No	Yes
W<1,0>	Yes	Yes	Yes	Yes
Bus Request (BR)				
BR<2>	No	Yes	No	Yes
BR<1,0>	Yes	Yes	Yes	Yes
Total Lines Required	29	42	46	58

4.2.2.1 Data Line Group (D//DC)

The Data Group shall consist of the Data (D) lines and the Data Check (DC) lines. The Data Group is a set of bidirectional lines which shall transfer header, data and acknowledge information between the bus master and the slave(s). The Data Group (D//DC) lines shall also be used to resolve priority during a Vie sequence.

4.2.2.1.1 Data Lines - Type 16.

Type 16 modules and buses shall provide 16 Data lines, D<15..0>. D0 shall be the least significant and D15 the most significant line.

4.2.2.1.2 Data Lines - Type 32.

Type 32 modules and buses shall provide 32 Data lines, D<31..0>. D0 shall be the least significant and D31 the most significant line. Type 16 data transfers shall always use D<15..0>.

4.2.2.1.3 Error Protection for the Data Line Group.

The Data Line Group shall use even parity for Class ED operation and a modified Hamming Code for Class EC operation when there is a single source for the signals. When there may be multiple sources for the signals, as during vie cycles and during multiple-slave acknowledges, the Data Line Group shall use duplication for Class ED operation and triplication for Class EC operation.

4.2.2.1.3.1 Class ED Operation.

4.2.2.1.3.1.1 Single Source. During bus cycles in which a single source is specified for the Data lines, valid symbols for the Data Line Group shall have even parity.

4.2.2.1.3.1.1.1 Type 16. The module that sources the Data lines shall also source the Data Check line such that the set of symbols on D<15..0>//DC<0> satisfies P(D<15..0>//DC<0>) = 0.

4.2.2.1.3.1.1.2 Type 32. The module that sources the Data lines shall also source the Data Check lines such that the set of symbols on D<15..0>//DC<0> satisfies P(D<15..0>//DC<0>) = 0 and D<31..16>//DC<1> satisfies P(D<31..16//DC<1>) = 0.

4.2.2.1.3.1.2 Multiple Sources. During Vie and Multiple Slave Acknowledge Cycles, the Data and Data Check lines may have multiple sources. For those operations, modules shall post duplicate copies of the required symbols using Data lines D<15..8> and D<7..0> as duplicate line sets.

Usage of the Data lines for Vie and Multiple Slave Acknowledge Cycles is specified in "5.3.3.1 Vie Sequence." and "5.2.3.3.2 Multiple Slave Acknowledge.," respectively.

4.2.2.1.3.2 Class EC Operation.

4.2.2.1.3.2.1 Single Source. During bus cycles in which a single source is specified for the Data lines, valid symbols for the Data Line Group shall use modified Hamming encoding as specified below.

4.2.2.1.3.2.1.1 Type 16. The module that sources the Data lines shall also source the Data Check lines such that the set of symbols on D<15..0>//DC<5..0> satisfies:

P(DC<5>, D<15,14,13,12,11,10,9,8>) = 0

P(DC<4>, D<15,14,7,6,5,4,3,2>) = 0

P(DC<3>, D<13,12,11,7,6,5,1,0>) = 0

P(DC<2>, D<15,13,10,9,7,4,3,0>) = 0

P(DC<1>, D<12,10,8,6,4,2,1,0>) = 0

P(DC<0>, D<14,11,9,8,5,3,2,1>) = 0

4.2.2.1.3.2.1.2 Type 32. The module that sources the Data lines shall also source the Data Check lines such that the set of symbols on D<31..16>//DC<6..0> satisfies:

P(DC<6>, D<31,30,29,28,27,26,25,24,23,22,20,19,17,16>) = 0

P(DC<5>, D<31,30,29,28,27,15,14,13,12,11,10,9,8>) = 0

P(DC<4>, D<31,26,25,24,23,15,14,7,6,5,4,3,2>) = 0

P(DC<3>, D<30,26,22,21,20,19,13,12,11,7,6,5,1,0>) = 0

P(DC<2>, D<29,25,22,21,18,17,15,13,10,9,7,4,3,0>) = 0

P(DC<1>, D<28,24,20,18,17,16,12,10,8,6,4,2,1,0>) = 0

P(DC<0>, D<27,23,21,19,18,16,14,11,9,8,5,3,2,1>) = 0

4.2.2.1.3.2.2 Multiple Sources. During Vie and Multiple Slave Acknowledge Cycles the Data and Data Check lines may have multiple sources. For those operations, modules shall post triplicate copies of the required symbols using D<15..8>, D<7..0> and DC<7..0> as triplicate line sets.

Usage of the Data lines for Vie and Multiple Slave Acknowledge Cycles is specified in "5.3.3.1 Vie Sequence." and "5.2.3.3.2 Multiple Slave Acknowledge.," respectively.

4.2.2.2 Cycle Type Line Group (CT//CTC).

The Cycle Type Group shall consist of the Cycle Type (CT) and Cycle Type Check (CTC) lines. The Cycle Type Group is a set of lines onto which the bus master shall post symbols to indicate the current bus cycle type. The bus Cycle Types shall be encoded as shown in Table 4-2.

Table 4-2. PI-bus Cycle Types and Valid Symbols

Cycle Type	Abbreviation	Class ED Symbol CT<2..0>	Class ED Symbol CTC<0>	Class EC Symbol CT<2..0>	Class EC Symbol CTC<2..0>
Abort	AB	111	1	111	001
Acknowledge	A	011	0	011	110
Data	D	001	1	001	011
Header 0	H0	101	0	101	100
Header	H	010	1	010	101
Idle	I	000	0	000	000
Suspend	S	110	0	110	010
Vie	V	100	1	100	111

4.2.2.3 Acknowledge Line Set (AS).

The Acknowledge Set is a group of lines onto which the slave(s) or contenders shall post symbols to indicate synchronization or to signal uncorrectable detected errors. Valid symbols for the Acknowledge Set shall be as defined in Table 4-3.

Table 4-3. Acknowledge Line Valid Symbols

Response	Abbreviation	Class ED Code AS<3,2>	AS<1,0>	Class EC Code AS<5,4>	AS<3,2>	AS<1,0>
Acknowledge	ACK	10	10	10	10	10
Negative Ack	NAK	11	11	11	11	11
Not Selected	NS	00	00	00	00	00
Recognize	RCG	01	01	01	01	01

4.2.2.4 Wait (W) Lines.

The Wait lines shall be a set of redundant lines which the current bus master and slave(s) may assert to obtain extra non-transfer bus cycles to supply information to the bus or to accept information from the bus.

4.2.2.4.1 Class ED.

Class ED modules and buses shall provide two Wait lines, W<1,0>, that shall operate as redundant lines. Valid symbols for this case shall be W<1,0> = 00, no wait request, and W<1,0> = 11, wait requested.

4.2.2.4.2 Class EC.

Class EC modules and buses shall provide three Wait lines, W<2..0>, that shall operate as redundant lines. Valid symbols for this case shall be W<2..0> = 000, no wait request, and W<2..0> = 111, wait requested.

4.2.2.5 Bus Request (BR) Lines.

The Bus Request lines shall consist of a set of redundant lines which shall be asserted to request that the current bus master release the bus.

4.2.2.5.1 Class ED.

Class ED modules and buses shall provide two Bus Request lines, BR<1,0>, that shall operate as redundant lines. Valid symbols for this case shall be BR<1,0> = 00, no bus request, and BR<1,0> = 11, bus requested.

4.2.2.5.2 Class EC.

Class EC modules and buses shall provide three Bus Request lines, BR<2..0>, that shall operate as redundant lines. Valid symbols for this case shall be BR<2..0> = 000, no bus request, and BR<2..0> = 111, bus requested.

4.2.3 BUS CLOCK.

Bus Clock shall be a single phase clock. All bus timing shall be referenced to the high-to-low transition of the bus clock. The generation and distribution of Bus Clock is beyond the scope of this specification. However, the period of Bus Clock shall be selected to guarantee that bus timing constraints are satisfied for the bus delays and clock skews resulting from the backplane design.

4.2.4 MODULE IDENTIFICATION.

PI-bus modules shall have inputs for a set of lines that shall be hardwired on the PI-bus backplane to provide a unique module identification.

4.2.4.1 Module Identification Lines.

The set of 5 Module Identification (MID) lines shall be hardwired on the backplane and shall be used by the module as the module's physical identification code. The identification codes shall consist of an unsigned binary number in the range of 0-31, inclusive, encoded in MID<4..0>.

4.2.4.2 Module Identification Parity Line.

A Module Identification Parity (MIP) line shall be hardwired on the backplane such that MID<4..0>//MIP<0> satisfies P(MID<4..0>//MIP<0>) = 1.

4.3 ELECTRICAL REQUIREMENTS

Electrical characteristics for the PI-bus backplane and modules shall be as specified herein.

4.3.1 BACKPLANE REQUIREMENTS.

4.3.1.1 Bus Signal Line Characteristic Impedance.

PI-Bus signal lines shall have a characteristic impedance of not less than 20 ohms and not more than 50 ohms for all operating and module loading conditions.

4.3.1.2 Bus Signal Line Termination.

Signal lines shall be terminated at each end of the backplane to a circuit which is the Thevenin-equivalent of a terminating resistor in series with a voltage source of not less than +1.9 Volts nor more than +2.1 Volts. The value of the terminating resistance shall be between 30 and 40 ohms, inclusive.

4.3.1.3 Bus Signal Line Resistance.

The series resistance for backplane signal lines shall be limited such that the maximum voltage rise from any asserted module output to the terminating resistance at either end of the backplane is less than 100 millivolts.

4.3.1.4 Module Identification Line Resistance.

The resistance of the grounded MID and MIP lines with respect to the signal ground shall be less than 10 ohms.

4.3.1.5 Bus Clock Requirements.

4.3.1.5.1 Voltage Levels.

The low level voltage for Bus Clock shall be less than or equal to +0.55 volts. The high level voltage for Bus Clock shall be greater than or equal to +2.4 volts.

4.3.1.5.2 Rise And Fall Time.

The rise time (Tr) of the Bus Clock from 0.8 volts to 2.0 volts shall be less than 5 nanoseconds. The fall time (Tf) of the Bus Clock from 2.0 volts to 0.8 volts shall be less than 5 nanoseconds.

4.3.1.5.3 Duty Cycle.

The ratio of the Bus Clock high state duration to the bus clock period measured at 1.5 Volts shall not be less than 0.45 nor greater than 0.55.

4.3.2 MODULE REQUIREMENTS.

4.3.2.1 Bus Clock Requirements.

4.3.2.1.1 DC Requirements.

4.3.2.1.1.1 Input Capacitance. Bus Clock capacitance to logic ground shall be less than 22 picofarads.

4.3.2.1.1.2 Input Inductance. Bus Clock series inductance from the module input to the receiver of the signal shall be less than 27 nanohenries.

4.3.2.1.1.3 Bus Clock Current. The maximum current sourced by the module when the clock input voltage is +0.55 volts shall be 1.6 milliamps. The maximum current into the module when the Bus Clock voltage is +2.4 volts shall be less than 100 microamps.

4.3.2.1.1.4 High-level Input Voltage. An Bus Clock input voltage of +2.0 volts or more shall be interpreted as a high level.

4.3.2.1.1.5 Low-level Input Voltage. A Bus Clock input voltage of +0.8 volts or less shall be interpreted as a low level.

4.3.2.1.2 AC Requirements.

Modules shall operate correctly with the Bus Clock characteristics specified in "4.3.1.5 Bus Clock Requirements.."

The maximum Bus Clock frequency for the module shall be specified. The minimum Bus Clock frequency shall be zero Hertz.

All PI-bus timing shall be referenced to the high-to-low transition of Bus Clock through a voltage of 1.5 volts.

4.3.2.2 Signal Line Requirements.

4.3.2.2.1 DC Requirements.

4.3.2.2.1.1 Input Capacitance. Signal line capacitance to logic ground shall be less than 22 picofarads.

4.3.2.2.1.2 Input Inductance. Signal line series inductance from the module input to the driver or receiver of the signal shall be less than 27 nanohenries.

4.3.2.2.1.3 Leakage Current. Over the input voltage range of +0.3 volts to +2.1 volts, the absolute value of the output current for any signal line which is not being asserted by the module shall be less than 100 microamps.

4.3.2.2.1.4 Low-level Sink Current. The low-level output sink current (Iol) drive capability for signal lines shall be greater than 95 milliamps at an output voltage of 1.15 volts.

4.3.2.2.1.5 High-level Output Voltage. The high-level output voltage shall be determined by the backplane signal line termination voltage which is +1.9 to +2.1 volts. The signal line outputs shall permit wired-OR operations on the bus.

4.3.2.2.1.6 Low-level Output Voltage. The low-level output voltage (Vol) for signal lines shall be less than 1.15 volts at an input current of 95 milliamps.

4.3.2.2.1.7 High-level Input Voltage. A signal line input voltage (Vih) of +1.6 volts or more shall be interpreted as a logic 0. A signal line input which is not electrically connected to the backplane (i.e. an open line) shall be interpreted as a logic 0.

4.3.2.2.1.8 Low-level Input Voltage. A signal line input voltage (Vil) of +1.45 volts or less shall be interpreted as a logic 1.

4.3.2.2.2 AC Requirements.

4.3.2.2.2.1 Signal Line Inputs. Figure 4-1 illustrates the timing relationships specified below.

4.3.2.2.2.1.1 Set-up Time. The maximum time that each input signal is required to be uniquely above or below the input voltage threshold for a logic 0 or logic 1 prior to the high-to-low transition of the clock (set-up time, Ts) shall be specified.

4.3.2.2.2.1.2 Hold-Time. The maximum time that each input signal is required to be uniquely above or below the input voltage threshold for a logic 0 or logic 1 following the high-to-low transition of the clock (hold time, Th) shall be specified and shall not exceed the minimum propagation delay time of the module.

4.3.2.2.2.1.3 Noise Rejection. The input signal lines shall reject and the Bus Interface shall not respond to any signal pulse whose width as measured between 1.5 volts on the low-to-high transition and 1.5 volts on the high-to-low transition is less than 4 nanoseconds.

4.3.2.2.2.2 Signal Line Outputs. The following specifications shall apply when the signal line is connected to the test circuit of Figure 4-2.

4.3.2.2.2.2.1 Propagation Delay. Propagation delay shall be measured with respect to the high-to-low transition of Bus Clock as illustrated in Figure 4-3. The reference clock voltage for timing shall be +1.5 volts. The reference signal voltage for timing shall be +1.5 volts.

The minimum and the maximum propagation delay (Tpdlh) for an output sig-

nal changing from a logic 1 (low voltage) to a logic 0 (high voltage) shall be specified for each output signal line.

The minimum and the maximum propagation delay (Tpdhl) for an output signal changing from a logic 0 (high voltage) to a logic 1 (low voltage) shall be specified for each output signal line.

4.3.2.2.2.2.2 Rise And Fall Time. The rise time (Tr) of an output signal from +1.2 volts to +1.8 volts shall be less than 9 nanoseconds. The fall time (Tf) of an output signal from +1.8 volts to +1.2 volts shall be less than 9 nanoseconds.

4.3.2.3 MID And MIP Lines.

A binary 1 shall be represented by a connection to signal ground and a binary 0 shall be represented by an open circuit. Modules shall incorporate any circuits they require to sense the MID and MIP lines. The absolute current into a grounded MID or MIP line shall be less than 1 milliamp. The maximum voltage that shall exist on an open MID or MIP line shall not exceed 25 volts.

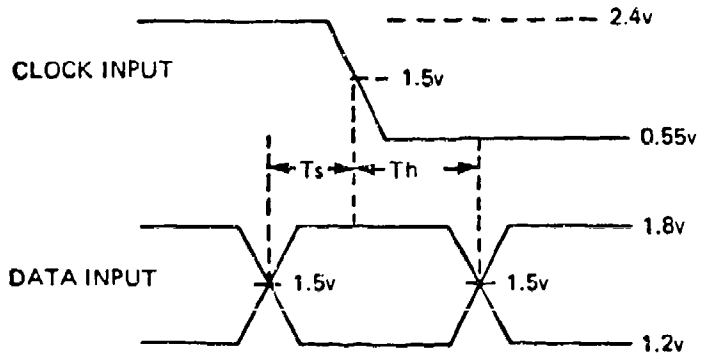


Figure 4-1. Set-up And Hold Timing

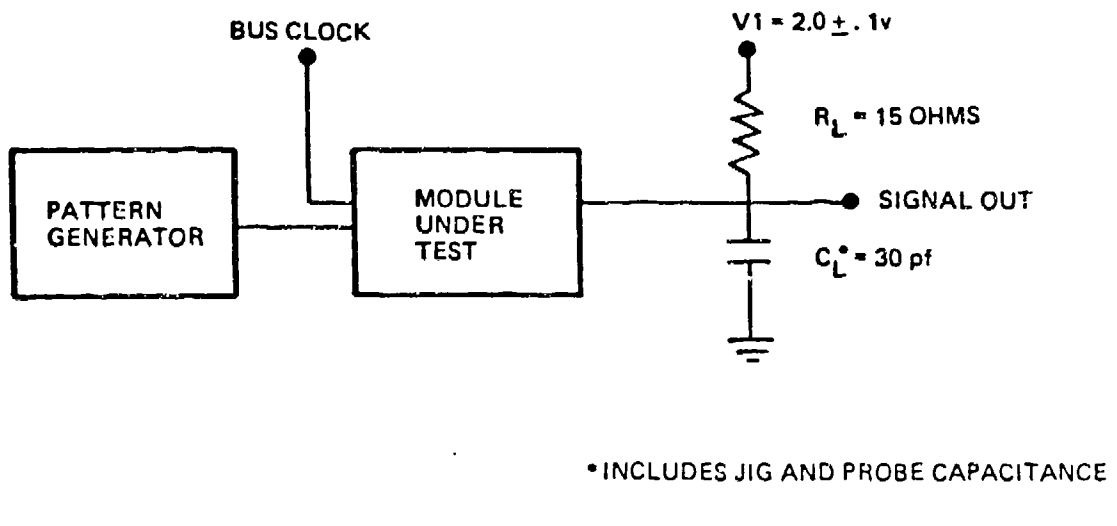


Figure 4-2. Signal Output Test Circuit

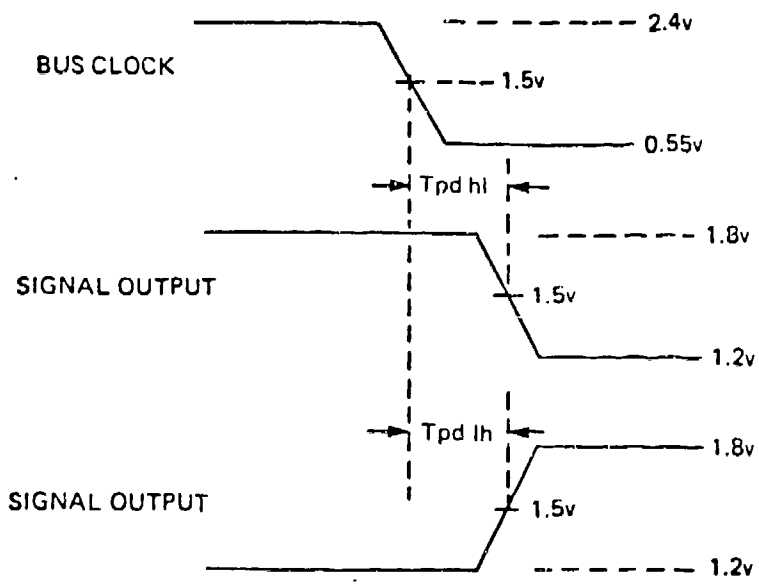


Figure 4-3. Output Signal Timing

Section 5

DATA LINK LAYER

5.1 INTRODUCTION.

The Data Link layer of the PI-bus is specified herein. The general protocol used by the PI-bus is defined through specification of the protocol state transitions and the generic message sequence. Detailed requirements for the protocol and communications sequences are specified by defining each sequence and the rules associated with the PI-bus protocol. Responses to exception conditions are defined.

5.2 GENERAL REQUIREMENTS.

5.2.1 INTRODUCTION.

The PI-bus uses a master-slave protocol under which communications sequences are defined for 1) transferring messages between modules and 2) changing bus mastership. The PI-bus communications sequences are listed in Table 5-1. The Vie sequence shall be performed only when there is no current bus master. All other sequences shall be performed under the control of the current bus master.

The PI-bus uses a set of protocol state transitions to define and control the communication sequences. Protocol state transitions shall be signaled on the Cycle Type (CT) lines and shall be controlled by the bus master. The slave(s) shall operate in synchronization with the bus master and shall signal compliance with protocol state transitions using the Acknowledge Set (AS) lines. Slave(s) shall also use the AS lines to notify the bus master of any uncorrectable errors that are detected.

The seven sequence states defined for the PI-bus protocol are summarized in Table 5-2. Within each sequence state, bus states are defined to distinguish individual bus cycles. The specific sequences of bus states required to perform PI-bus communications are defined under "5.3 DETAILED REQUIREMENTS.." In this section, general requirements for the overall operation of the PI-bus are specified by reference to the sequence states and the generic message protocol they support.

Table 5-1. PI-bus Communications Sequences

Sequence Type	Function
Mastership Sequences:	
Vie	Assigns bus mastership to the highest priority module contending for mastership through arbitration.
Tenure Pass Message	Transfers bus mastership from current bus master to another module or changes the bus master's message priority.
Message Sequences:	
Parameter Write	Transfers a 1 word parameter and a 32 bit address from the bus master device to the slave device(s).
Block Message	Transfers up to 65,536 datum units from slave device to master device or from master device to slave(s). Master sends a 32 bit address and may send 6 other Header words. May be used to continue a suspended message.
Bus Interface Message	Transfers up to 256 words from slave bus interface to the master device or from master device to slave Bus Interface(s). Master provides an 8 bit address.
Exception Sequences:	
Suspend	Suspends a Block Message data sequence and transfers Resume Control Words from the slave to the master.
Abort	Abnormally terminates current sequence.

Table 5-2. PI-bus Protocol States

Protocol State	Function
Idle	Bus not in use and no current bus master defined.
Vie	State following Idle. Used to select the next bus master from one or more contending modules. The module with the highest priority is selected as the next bus master.
Header	State in which information is transmitted by the master to specify the type of message sequence, identify modules to participate as slaves and specify additional application dependent information.
Header Acknowledge (Header Ack)	State following Header during which slave module(s) provide sequence status information to the master.
Data	State during which data are transferred between the slave module(s) and the master for Block Messages and Bus Interface Messages. Block Message Suspend sequences are performed under this protocol state.
Data Acknowledge (Data Ack)	State following Data during which slave module(s) provide message status information to the master.
Abort	State used to abnormally terminate another bus sequence.

5.2.2 PROTOCOL STATE TRANSITIONS.

The protocol states which shall be used in PI-bus operations are illustrated in Figure 5-1. All state transitions shall occur on the high-to-low transition of Bus Clock. The allowable transitions between protocol states are specified in the sections below and in Figure 5-1.

5.2.2.1 Idle.

The bus shall enter the Idle state whenever all Cycle Type lines are released. There shall be no bus master during Idle and the current bus master priority code shall be undefined. Idle shall consist of two or more consecutive bus cycles in which the Cycle Type lines are released. No PI-bus operations shall be performed during Idle except that the symbol NAK (Negative Acknowledgement) may be posted on the AS lines as specified in "5.2.3.1.2.2 Uncorrectable Errors." Vie shall be the only valid successor state to Idle. The Idle state shall be terminated and the Vie state entered only when one or more modules post the symbol V on the Cycle Type lines.

5.2.2.2 Vie.

The Vie state shall consist of eight bus cycles which shall be used to select the next bus master from one or more contenders. The Vie state shall be succeeded by the Header state except that if no bus master is selected due to erroneous operation, the bus shall return to the Idle state.

5.2.2.3 Header.

A bus master's tenure shall begin when the Header state is entered from the Vie state or from the Header Acknowledge state of the Tenure Pass message. The current bus master's tenure shall continue when the Header state is entered from the Header Acknowledge state of the Parameter Write sequence, from the Data Acknowledge state or from the Abort state. During the Header state, the bus master shall transmit header information across the bus on two or more bus cycles.

The Header shall specify the type of message sequence to be performed, identify the modules required to participate in the sequence as slaves and define the number of data transfer cycles required for the sequence. The Header state shall be succeeded by the Header Acknowledge state except that Abort may be entered to terminate the sequence.

5.2.2.4 Header Acknowledge.

The Header Acknowledge state shall be used to transmit message status from the slave module(s) to the master. The transitions out of the Header Acknowledge state shall be as specified below:

1. For a Parameter Write message sequence, the successor states to Header Acknowledge shall be Header, Idle and Abort. A transition to Header shall initiate a new message and extend the the current bus master's ten-

ure. A transition to Idle shall terminate the current bus master's tenure. Abort may be entered to terminate the Parameter Write message.

2. For Block Message and Bus Interface Message sequences, the successor states to Header Acknowledge shall be Data and Abort. A transition to Data continues the current bus master's tenure. Abort may be entered to terminate the message.

3. For a Tenure Pass Message sequence, the successor states shall be Header or Abort except that when the intended next bus master does not require or fails to acquire bus mastership, the successor state shall be Idle. The current bus master's tenure shall end at the conclusion of a Tenure Pass Message Header Acknowledge (HAZ) and the new bus master's tenure shall begin on the next cycle with entry into the Header state. Abort may be entered from a Tenure Pass Message except on the last cycle (cycle HAZ) of the message.

5.2.2.5 Data.

The Data state shall consist of a sequence of Data transfer cycles performed as part of a Block Message or Bus Interface Message. Data may be transferred from the master to the slave(s), defined as a write sequence, or from the slave to the master, defined as a read sequence. For Block Message sequences only, the Data sequence may be suspended by entry into the Suspend state. Unless a Data sequence is suspended or terminated by entering Abort, the successor state to Data shall be Data Acknowledge.

5.2.2.6 Suspend.

The Suspend state shall be used to signal the pending interruption of a Block Message Data sequence as specified in the detailed requirements (see "5.3.5.1 Suspend."). A suspended Block Message Data sequence can be resumed by another Block Message whose header contains the appropriate Resume Control Words. The successor state to Suspend shall be Data Acknowledge except that the sequence may be terminated by entering Abort.

5.2.2.7 Data Acknowledge.

The Data Acknowledge state shall be used to transfer acknowledge information from the slave(s) to the master during a Block Message or Bus Interface Message sequence. The successor states to Data Acknowledge are Header, Idle and Abort. A transition to Header shall initiate a new message and extend the the current bus master's tenure. A transition to Idle shall terminate the current bus master's tenure. Abort may be entered to terminate a message.

5.2.2.8 Abort.

The Abort state shall consist of four consecutive bus cycles in which the Abort cycle type is posted on the CT lines. The successor states to Abort shall be Header and Idle. A transition to Header shall initiate a new message and extend the the current bus master's tenure. A transition to Idle shall

terminate the current bus master's tenure.

5.2.2.9 Tenure Limitations.

5.2.2.9.1 Bus Request To Vie Interval.

When Bus Request is asserted, the bus master shall limit the number of bus cycles remaining in the current tenure to the sum of the bus cycles specified by the contents of the Vie Interval A Register plus the contents of the Vie Interval B Register plus six cycles (see "5.3.7.3.4 Vie Interval A Register - Address 3." and "5.3.7.3.5 Vie Interval B Register - Address 4."). Section "5.3.3.3 Bus Request." specifies procedures which the bus master shall use to relinquish tenure and permit a Vie sequence in response to Bus Request.

5.2.2.9.2 Absolute Tenure Limit.

PI-bus modules shall internally limit each of their individual tenures as bus master to a maximum of (2**24)+8 bus cycles. The cycle count shall begin with the first H0 cycle of the master's tenure and shall include all bus cycles (including non-transfer cycles). Each module shall provide a hardwired mechanism to automatically force all signal outputs from the module to end tenure such that this tenure limit is not exceeded. The module may resume normal operation, including vying for the bus, after allowing the bus to be in the Idle state for a minimum of two cycles.

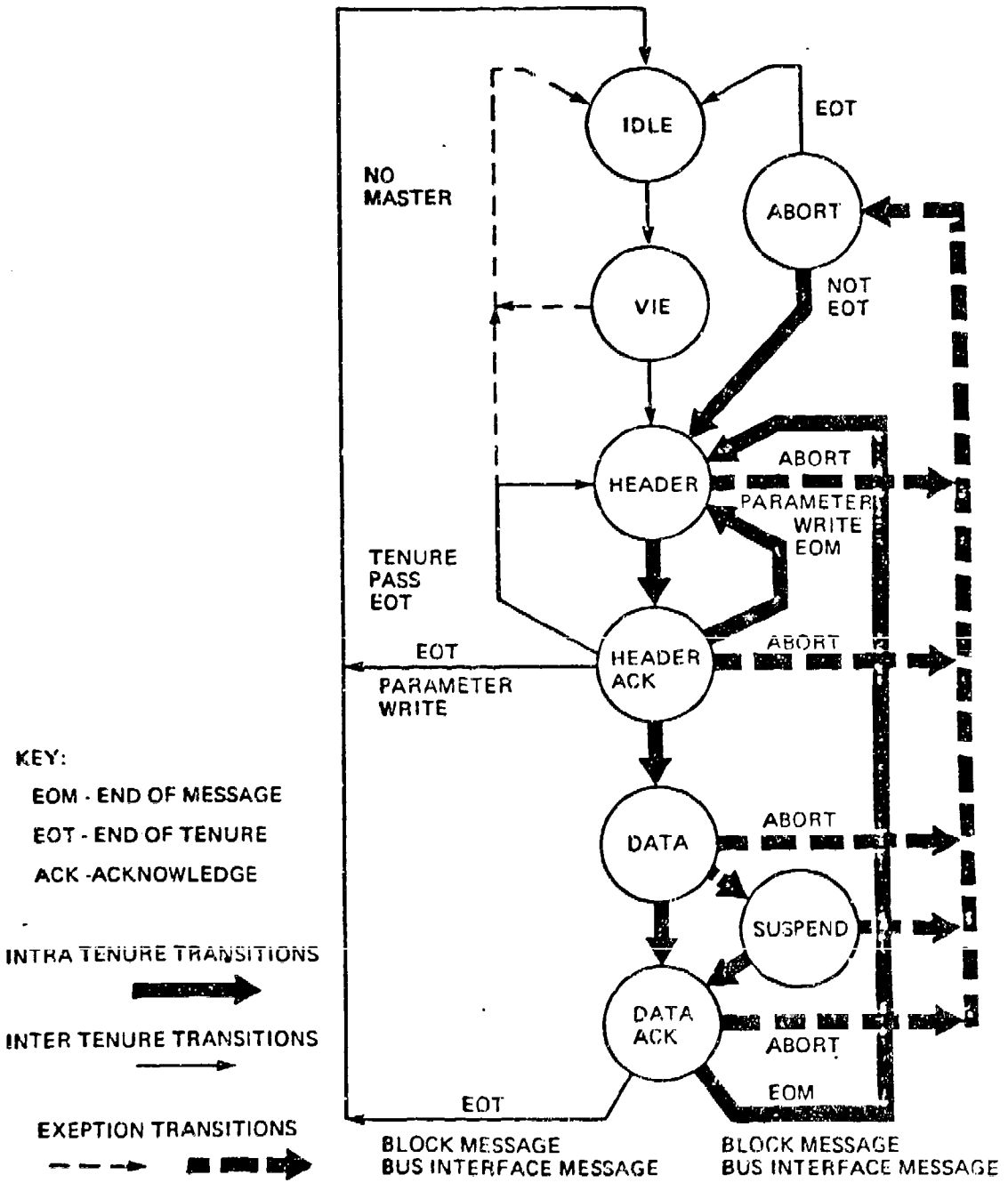


Figure 5-1. PI-bus Protocol State Diagram

5.2.3 GENERIC MESSAGE.

The generic message sequence that forms the basis for the PI-bus message sequences is described in this section. The Vie sequence is specified in "5.3.3 Bus Mastership." and the exception sequences are specified in "5.3.5 Exception Sequences.."

Table 5-3 illustrates the generic PI-bus message sequence which shall be composed of Header, Header Acknowledge, Data and Data Acknowledge sequences that correspond to the protocol states described in the preceding section.

5.2.3.1 Generic Message Sequence.

5.2.3.1.1 Normal Operation.

The Data (D) lines transfer information between the master and slave modules to accomplish the following:

1. signal the type of message sequence to be performed;
2. establish a communications path to the slave module(s);
3. transfer data between the master and the slave(s); and
4. transfer acknowledge information from the slave(s) to the master.

The bus master shall use Type 32 message sequences only when the master and all modules selected as slaves for that sequence are operating as Type 32 modules on a Type 32 bus. For a Type 32 bus, Type 32/Type 16 message sequence selection can be made on a message-by-message basis and thus may vary during the bus master's tenure.

The Cycle Type (CT) and Acknowledge Set (AS) lines shall provide handshaking between the master and slave(s) to control the sequence of bus states. The AS lines shall also be used by the slave(s) to report errors.

5.2.3.1.1.1 Header. The bus master shall initiate a message sequence by transmitting Header information on the D lines. The bus master shall post the symbol H0 on the Cycle Type (CT) lines during the first bus cycle of Header transfer and shall post H for each of the remaining bus cycles of header transfer.

The header shall specify the module(s) which are selected as slave(s) for the message sequence. Active module(s) which are addressed by the slave ID field of HWA shall become slaves on the third cycle of Header transfer. Slaves shall signal their participation in the message by posting the symbol RCG (Recognize) on the Acknowledge Set (AS) lines beginning with the third cycle of header transfer and continuing until header transfer is complete. Modules shall cease being slaves when the current message is closed by a normal completion, Abort or Suspend. All modules shall ensure that the AS lines are released during the first two cycles of Header except that NAK shall be

Table 5-3. Generic PI-bus Message Sequence

	PROTOCOL BUS STATE					
SIGNAL LINES	HEADER			HEADER ACKNOWLEDGE	DATA	DATA ACKNOWLEDGE
DATA Source=	Header Master			Acknowledge Slave(s)	Data Master (Write) Slave (Read)	Acknowledge Slave(s)
CYCLE TYPE Source=Master	H0	H		A	D (S)	A
ACKNOWLEDGE Source=	NS	NS	RCG Slave	ACK Slave	RCG Slave	ACK Slave

asserted as specified in "5.2.3.1.2.2 Uncorrectable Errors" to report errors from the preceding sequence.

5.2.3.1.1.2 Header Acknowledge. The Header Acknowledge sequence shall follow the Header sequence. A single Header Acknowledge transfer cycle shall be used for all single slave sequences. The Tenure Pass Message sequence shall include an additional (non-transfer) cycle to ensure a proper transition of mastership. The bus master shall post the Header Acknowledge Cycle (A) symbol on the CT lines during the Header Acknowledge cycle in which the slave is scheduled to post the slave Acknowledge word. The slave shall indicate synchronization with the bus master by posting ACK on the AS lines during the Header Acknowledge cycle. Five Header Acknowledge transfer cycles shall be used for a multiple slave sequence. During the first multiple slave header acknowledge cycle, all slaves post a message status symbol on the data lines and the ACK symbol on the AS lines. During each of the four remaining acknowledge cycles, eight of the thirty-two modules are assigned a bit position on the data lines upon which to post an acknowledge bit and shall indicate synchronization with the bus master by posting ACK on the AS lines.

The Header Acknowledge sequence shall complete the Tenure Pass Message and Parameter Write Message. The Block Message and Bus Interface Message sequences shall continue with a Data sequence consisting of one or more data transfer bus cycles.

5.2.3.1.1.3 Data. The bus master shall post the symbol D during each cycle of the Data sequence. The slave module(s) shall post RCG during each cycle of the Data sequence. The bus master shall transmit data during write sequences and the slave shall transmit data during the read sequences.

5.2.3.1.1.4 Data Acknowledge.. The Data sequence shall be followed by a Data Acknowledge sequence. The Data Acknowledge sequence is identical in form to the Header Acknowledge sequence. Block Messages and Bus Interface Messages shall be concluded at the end of the Data Acknowledge sequence.

5.2.3.1.2 Operation Under Exception Conditions.

5.2.3.1.2.1 Block Message Suspend. The Data sequence of a Block Message may be suspended by the bus master to permit higher priority communications. The Suspend sequence shall be performed as defined in "5.3.5.1 Suspend.."

5.2.3.1.2.2 Uncorrectable Errors. Modules which are slaves shall signal uncorrectable detected errors by posting the symbol NAK on the AS lines and providing an error log in the Acknowledge words as specified herein. Modules shall post the symbol NAK in response to an uncorrectable error which occurs during a Vie or during a Tenure Pass Message sequence.

A module that detects an uncorrectable error which applies to the operation of bus cycle N shall post the symbol NAK on bus cycle N+2. If the detected error occurred during the last two cycles of a message, the resultant NAK occurs during the first two cycles of the following message or Idle. Modules which are not slaves nor contenders in a particular message sequence shall not otherwise post NAK during that sequence.

Slave modules shall record detected error conditions for the current message in the single slave Acknowledge word or Multicast Acknowledge Register as appropriate. The Bus Interface should also notify the device of any detected errors. When an Acknowledge Word is transmitted on the bus or stored into the Multicast Acknowledge Register, the error field shall not include errors which apply to the immediately preceding bus cycle.

NAK shall have no specified effect on the resulting operation of the PI-bus other than that when NAK occurs or the asserted state of an Acknowledge word error bit is detected, the master Bus Interface should report that fact to the master device. The protocol provides the Abort sequence as a means for the message to be terminated if required by the device.

5.2.3.2 Generic Header Definition.

The PI-bus protocol defines message headers consisting of two to ten words. The first header word, Header Word A (HWA), shall be used in all message sequences to define 1) the type of message and 2) the slave modules for that message. The number of scheduled cycles in a message shall be defined by the message type and a datum transfer cycle count which shall be given implicitly in HWA or explicitly in the second header word, Header Word B (HWB). For the Bus Interface Message, HWB shall also contain an eight bit virtual address

for the Bus Interface registers. The 32 bit virtual address used in Block and Parameter Write messages shall be contained in the third and fourth header words, Header Word C0 (HWC0) and Header Word C1 (HWC1). Block Message-extended header sequences provide six additional header words, HWD0 through HWD5, for application dependent uses.

The generic format for Header Word A (HWA) is defined in this section. Formats for the remaining Header words are specific to each message sequence and are defined in "5.3 DETAILED REQUIREMENTS.."

5.2.3.2.1 Header Word A.

The format illustrated in Figure 5-2 shall be used for Header Word A. The fields of HWA shall be as specified below.

5.2.3.2.1.1 Slave Identification (ID) Field. The Slave ID field of HWA shall specify the modules required to participate in the message sequence as slaves. The Slave ID field shall provide an eight bit virtual slave address. The virtual slave address (Slave ID) range shall be partitioned into 32 single slave physical addresses, a slave broadcast address and 223 optional logical slave addresses. The broadcast Slave ID shall select all active modules as slaves, including the bus master module.

The logical Slave ID's shall be used only to define aliases for individual slave physical addresses or sets of slave physical addresses. The use of a logical Slave ID rather than a physical Slave ID in addressing a module shall not elicit any slave module response other than that which would have been produced by using the module's physical Slave ID. A Slave ID value of 0 to 31 shall specify the single module whose Module Identification matches the Slave ID field. A Slave ID value of 32 shall select all active modules as slaves. The Slave ID values 33 to 255 shall select any active modules with the given Slave ID enabled. The number of modules which respond to each of the logical Slave ID values 33 to 255 is system dependent. This means that Slave ID values 33 - 255 can be used for single slave messages or multiple slave messages depending on the application.

During message sequences where the current bus master module is selected as a slave, the module shall act as both a master and a slave to the extent that the complete standard bus message sequence can be observed on the bus lines.

5.2.3.2.1.2 Format Field (F). The Format field shall specify whether the message sequence will be performed using 16 bit (Type 16) transfers or 32 bit (Type 32) transfers. The master device must insure that 32 bit transfers are used only when no Type 16 module is selected as a slave.

5.2.3.2.1.3 Message Type Field (MSG TYPE). The Message Type field shall specify the type of message sequence to be performed according to the values in Table 5-4. The master device must ensure that only the multiple slave Message Types are used when more than one module is selected by the Slave ID field.

5.2.3.2.1.4 Access Type Field (AT). The Access Type (AT) field shall be passed from the master device to the slave module for use as defined herein and listed in Table 5-5.

Application specific AT codes may be used to classify the type of device access to be performed during a particular message. Typical uses are 1) to specify direct or indirect addressing, 2) to specify address increments, 3) to specify device dependent interpretations for extended header words and 4) to access specific processes in the device.

For Block Messages, bit 13 shall be used to signal the device that the current message is a new Block Message (bit 13 = 0) or a resumption of a previously suspended Block Message (bit 13 = 1).

For Bus Interface Messages, the code 000 shall be used to access the Data Link address space. The Bus Interface physical and Data Link layers shall not utilize any AT field information except that contained in a Bus Interface Message. Codes 001 through 011 shall be reserved for future use by higher level protocols. Higher level protocols, such as those which provide communications between modules on different backplanes, are beyond the scope of this specification.

Reserved AT codes shall be defined only by future versions of this specification.

Figure 5-2. Header Word A Format - Data Lines

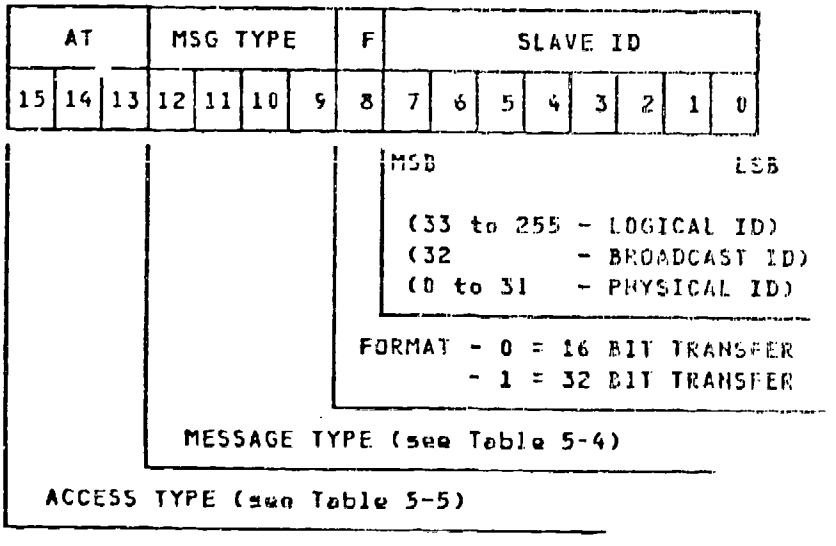


Table 5-4. Message Type Codes

MESSAGE TYPE	SINGLE OR MULTIPLE SLAVES	READ OR WRITE	MESSAGE TYPE CODE MWA <12..9>
PARAMETER WRITE	SINGLE	WRITE	0 0 0 1
	MULTIPLE	WRITE	0 0 1 1
BLOCK MESSAGE			
-SHORT HEADER	SINGLE	WRITE	0 1 0 1
	SINGLE	READ	0 1 0 0
	MULTIPLE	WRITE	0 1 1 1
-EXTENDED HEADER	SINGLE	WRITE	1 1 0 1
	SINGLE	READ	1 1 0 0
	MULTIPLE	WRITE	1 1 1 1
TENURE PASS	SINGLE	WRITE	1 0 1 0
BUS INTERFACE	SINGLE	WRITE	1 0 0 1
	SINGLE	READ	1 0 0 0
	MULTIPLE	WRITE	1 0 1 1

NOTE: Codes not listed above are reserved.

Table 5-5. Access Type Codes

SEQUENCE TYPE	ACCESS TYPE CODE HWA <15..13>	
PARAMETER WRITE	000 THRU 110	- APPLICATION SPECIFIC. (PASSED TO DEVICE)
	111	- RESERVED.
BLOCK MESSAGE		
BITS 15-14	00 THRU 11	- APPLICATION SPECIFIC (PASSED TO DEVICE)
BIT 13	0	- NEW MESSAGE
	1	- RESUME PREVIOUS MESSAGE
TENURE PASS	000	- TENURE PASS.
	001 THRU 111	- RESERVED.
BUS INTERFACE	000	- BUS INTERFACE LINK REGISTER SPACE.
	001 THRU 011	- RESERVED FOR HIGHER LEVEL PROTOCOL ACCESS.
	100 THRU 110	- IMPLEMENTATION DEFINED REGISTER SPACE.
	111	- RESERVED.

5.2.3.3 Header And Data Sequence Acknowledgement.

Header and Data Acknowledge sequences have the same form and use the same word formats. There are two basic formats for the acknowledgement, single slave and multiple slave. The single slave Acknowledge sequence shall be used when the Sequence Type field in HWA specifies a single slave sequence and the multiple slave Acknowledge sequence shall be used whenever the Sequence Type field in HWA specifies a multiple slave sequence. The single slave Acknowledge sequence transfers one word of message status information from the slave to the master. The multiple slave Acknowledge sequence transfers 1) eight bits of aggregate message status information from the slaves to the master and 2) an individual bit of message status information from each of the 32 possible slave devices to the master.

5.2.3.3.1 Single Slave Acknowledge.

The slave module shall perform error checking and logging during the message sequence. The Single Slave Acknowledge Word (AWS) defined herein provides the master with a record of the logged errors and the Module Identification of the slave.

The Single Slave Acknowledge Word shall be transmitted from the slave to the master during the Header and Data Acknowledge cycles of each single slave sequence. The master Bus Interface should pass the Single Slave Acknowledge Word to the master device.

The Single Slave Acknowledge Word format shall be as shown in Figure 5-3 and specified below.

5.2.3.3.1.1 Slave Module Identification Field (MID). The Slave Module Identification field shall contain the MID for the slave.

5.2.3.3.1.2 Acknowledge Word Type Field (AWT). The Acknowledge Word Type (AWT) field shall contain a two bit binary code as specified in Figure 5-3. An AWT code of 00 shall specify that the slave has closed the message sequence with this header or data acknowledge, as appropriate. In conjunction with an S field value of 0, an AWT of 00 shall specify message complete. In conjunction with an S field value of 1, an AWT of 00 shall specify that the Acknowledge Word completes the slave's response to a Suspend sequence as specified in "5.3.5.1 Suspend.." AWT codes 01, 10 and 11 shall specify that the slave is acknowledging the completion of the header sequence. The codes 01 and 10 further specify that the slave will respond to a Data sequence suspend with two or eight Resume Control Words, respectively. The S field code shall be 0 for an AWT code of 01 or 10. An AWT code of 11 shall further specify that the slave cannot perform a suspend sequence during the current message. The S field code shall be 1 for an AWT code of 11. The slave shall use an AWT code of 11 and an S code of 1 for any Bus Interface Message Data Acknowledge.

5.2.3.3.1.3 Errors Field. The slave module shall specify detected errors in bits 7 through 13 of the Acknowledge word. Errors reported in the Header Acknowledge Word shall be those errors that originate in the current message from the start of the Header through the second cycle prior to the Header Acknowledge. Errors reported in the Data Acknowledge Word shall be those errors that originate in the current message from one cycle prior to the Header Acknowledge through the second cycle prior to the Data Acknowledge. In addition to logging current message error indications, the Bus Interface should report detected errors to the device at the time of detection.

The definition of each error type in the single slave acknowledge word shall be as listed below:

Correctable Line Error	A signal line error has been detected and corrected.
Uncorrectable Line Error	A signal line error that cannot be corrected

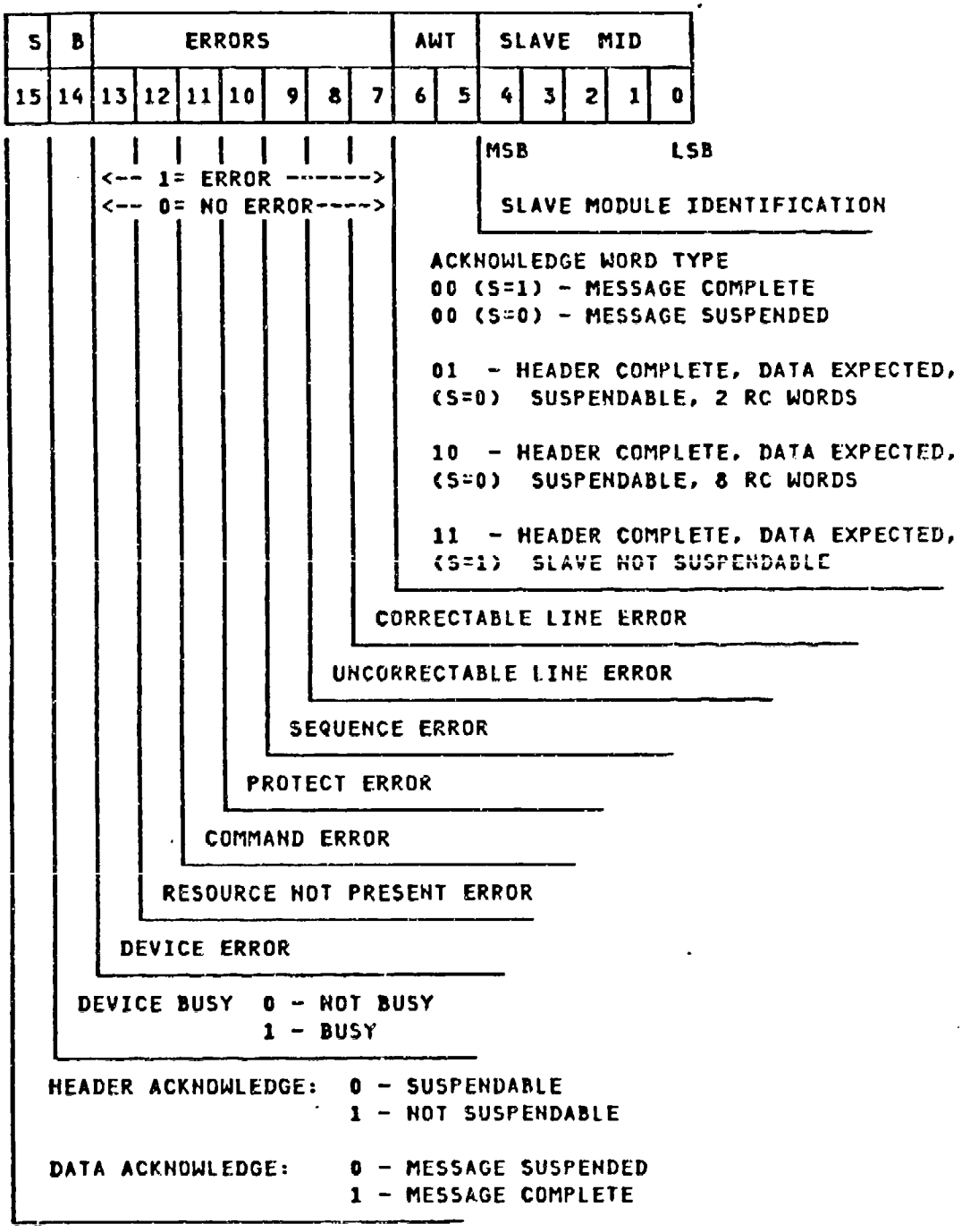
has been detected.

Sequence Error	The Cycle Type, Acknowledge Set, Wait and Bus Request line sequence of states is not in agreement with defined protocol sequences or rules.
Protect Error	A Bus Interface Message write operation has been attempted which is write protected.
Command Error	A Header Word A has been received which is not in agreement with the defined protocol or the slave is unable to perform the commanded operation because the current bus master's priority code is unknown.
Resource Not Present Error	A resource or capability that is not implemented has been addressed in this module.
Device Error	Module device has detected an error attempting to perform a bus related operation.

5.2.3.3.1.4 Busy Field (B). The slave module shall specify in bit 14 of the Acknowledge word whether the slave device is Busy or not Busy. The device shall be recorded as Busy only when the device is unable to accept an otherwise valid message because of other operations in progress. The master should abort any sequence in which the slave device is specified as Busy. The master may retry the message at a later time.

5.2.3.3.1.5 Suspend Field (S). The slave shall specify in bit 15 of the Header Acknowledge Word whether the message is suspendable or not suspendable. Only Block Messages may be specified as suspendable. The slave shall specify in bit 15 of the Data Acknowledge Word whether the Data Acknowledge is in response to a Suspend sequence or the completion of the message.

Figure 5-3. Single Slave Acknowledge Word Format - Data Lines



5.2.3.3.2 Multiple Slave Acknowledge.

A Multiple Slave Acknowledge sequence shall consist of one bus transfer cycle to transmit aggregate message status and four bus transfer cycles to transmit individual acknowledge bits. The first transfer cycle shall be used by each slave to transmit a Message Status Word to the master. The individual Message Status Words shall be wire-ORed on the bus to form an aggregrate Message Status Word. The remaining four transfer cycles shall be used to transmit multiple slave Acknowledge symbols from the slave(s) to the master. The five transfer cycles are labeled HA0, HA1, HA2, HA3 and HA4 for a Multiple Slave Header Acknowledge sequence or DA0, DA1, DA2, DA3 and DA4 for a Multiple Slave Data Acknowledge sequence.

During cycles HA0 and DA0, slave modules shall transmit a Message Status word to the master using Data lines <15...0>. The format of the Message Status word shall be as defined in Figure 5-4 and Figure 5-5. Data lines <15...8> shall have the same meaning as bits <15...8> of the Single Slave Acknowledge Word and these bits shall be replicated on Data lines <7...0>, respectively, to permit error checking. If there is an uncorrectable error in the Data line group, the module shall assume that both lines in any affected pair of redundant lines are asserted. For Class EC messages, bits <15...8> shall also be replicated on Data Check lines <7...0>, respectively, to permit error correction. Modules shall not assert any Data Group line on cycle HA0 or DA0 other than the lines defined above. The Master Bus Interface should pass the aggregate Message Status to the master device.

Header acknowledge cycles HA1 through HA4 shall be used to transfer "Acknowledge" multiple slave acknowledge symbols from the slave(s) to the master. Slave modules with MID values 0 through 7 shall post the "Acknowledge" symbols shown in Table 5-6 and Table 5-7 on the Data Group during cycle HA1. Slave modules with MID values 8 through 15 shall post their "Acknowledge" symbol during cycle HA2 of the sequence. Slave modules with MID values 16 through 23 shall post their "Acknowledge" symbol during cycle HA3 of the sequence and slave modules with MID values 24 through 31 shall post their "Acknowledge" symbol during cycle HA4 of the sequence. Modules shall not assert any Data Group line other than the lines included in their symbol on their assigned Acknowledge cycle.

Data acknowledge cycles DA1 through DA4 shall be used to transfer the "Acknowledge" or "No Acknowledge" multiple slave acknowledge symbols defined in Table 5-6 and Table 5-7 from the slave(s) to the master. The "Acknowledge" multiple slave Acknowledge symbol shall be posted on the Data Group during the assigned cycle of a Data Acknowledge sequence when the module is a slave and has detected no uncorrectable errors in the current sequence. Otherwise the slave shall post "No Acknowledge" during the assigned Acknowledge cycle. Slave modules with MID values 0 through 7 shall post their multiple slave Acknowledge symbol on the Data Group during cycle DA1. Slave modules with MID values 8 through 15 shall post their multiple salve Acknowledge symbol during cycle DA2 of the sequence. Slave modules with MID values 16 through 23 shall post their multiple slave Acknowledge symbol during cycle DA3 of the sequence and slave modules with MID values 24 through 31 shall post their multiple

slave Acknowledge symbol during cycle DA4 of the sequence. Modules shall not assert any Data Group line other than the lines included in their symbol on their assigned Acknowledge cycle.

The multiple slave Acknowledge symbols posted by the slave(s) during a particular Acknowledge cycle shall be logically OR'ed on the bus to produce one of the four Multiple Slave Acknowledge Words (AWM1, AWM2, AWM3 or AWM4) which shall be used during each multiple slave acknowledge sequence. The master Bus Interface should pass the Multiple Slave Acknowledge Words to the master device.

In addition to posting their assigned acknowledge symbol, slave modules shall post the symbol ACK (or NAK if required in response to an error) on the AS lines during their assigned Acknowledge cycle. Slave modules shall post the symbol NS (or NAK if required in response to an error) on the AS lines during the Acknowledge cycles in which they are not assigned to post their Acknowledge symbol.

Errors shall be logged during multiple slave sequences as specified for single slave sequences. A multicast sequence acknowledge word which has the same format and information that a Single Slave Acknowledge Word would have had if the sequence had been a single slave sequence shall be stored in the Multicast Acknowledge Register (see "5.3.7.3.1 Multicast Acknowledge Register - Address 0.") on the slave's assigned Header Acknowledge cycle. During multiple slave Bus Interface Message and Block Message sequences, a word equivalent to the Data Acknowledge word for a single slave sequence shall be formed. The acknowledge information stored in Multicast Acknowledge register bits <14..7> on the Header Acknowledge cycle shall be logically OR'ed with bits <14..7> of the equivalent single slave Data Acknowledge word and the result stored in bits <14..7> of the Multicast Acknowledge register on the slave's assigned Data Acknowledge cycle. Bit <15> and bits <6..0> of the equivalent single slave Data Acknowledge word shall be stored in Multicast Acknowledge Register bit <15> and bits <6..0>, respectively, on the slave's assigned Data Acknowledge cycle. All detected errors should be reported to the device at the time of detection.

Figure 5-4. Multiple Slave Status Word Format (AWM0) - Data Lines

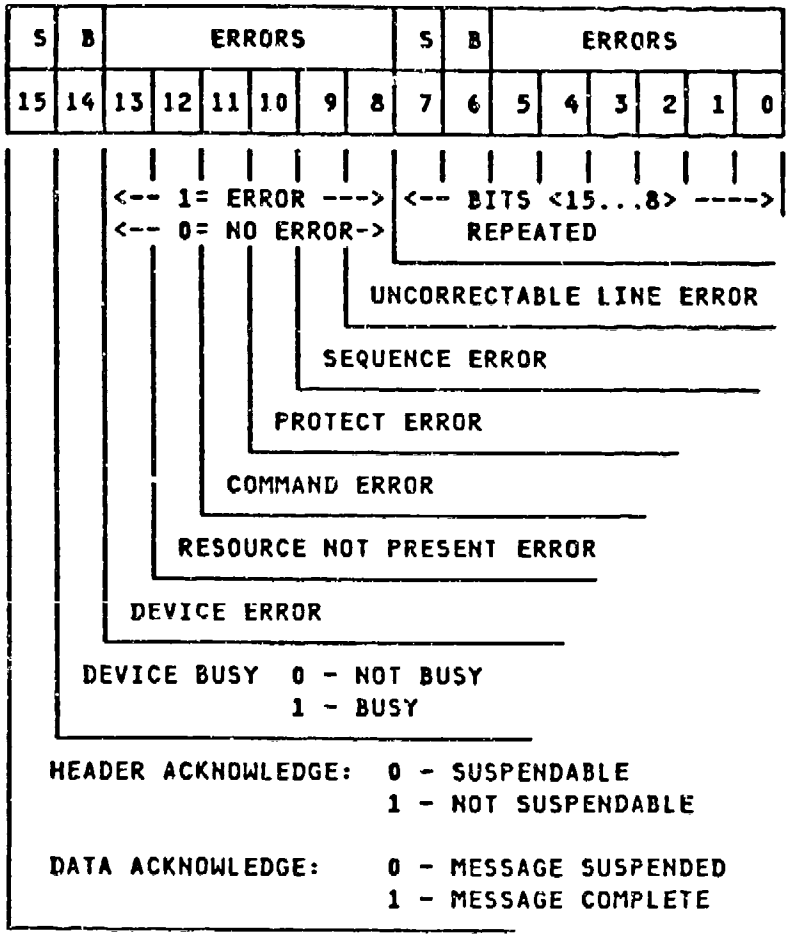


Figure 5-5. Multiple Slave Status Word Format - Data Check Lines

(Used only for Class EC)

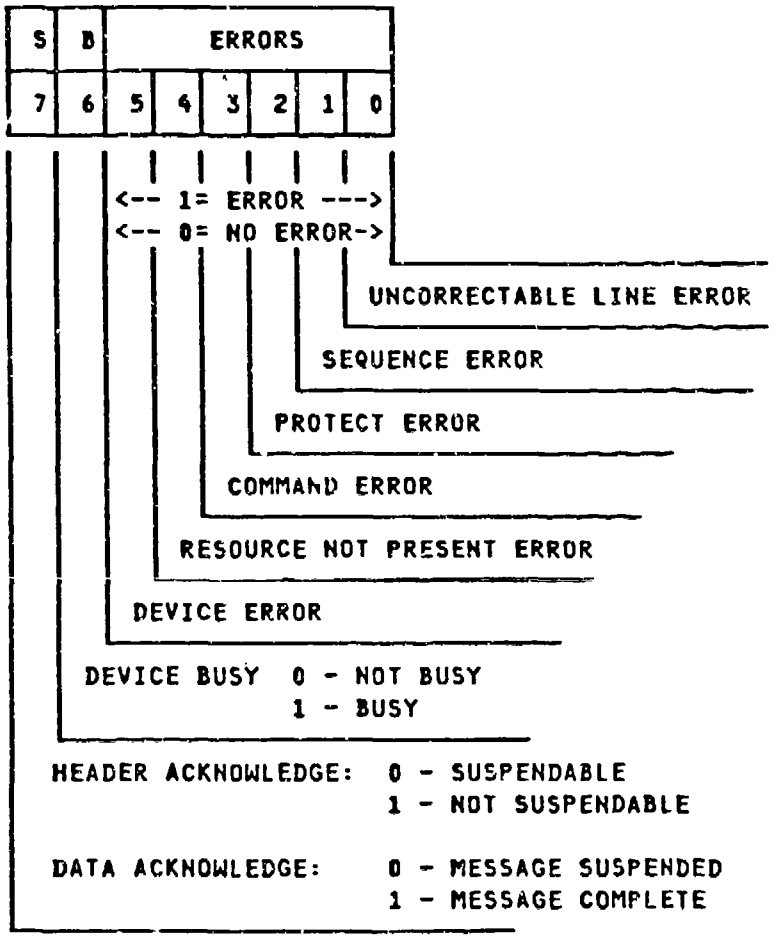


Table 5-6. Multiple Slave Acknowledge Symbol Formats (Four Words)

MODULE ID ASSIGNMENT				DATA LINES															
AWM1	AWM2	AWM3	AWM4	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	8	16	24	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0	A
1	9	17	25	0	0	0	0	0	0	A	0	0	0	0	0	0	0	A	0
2	10	18	26	0	0	0	0	0	A	0	0	0	0	0	0	0	A	0	0
3	11	19	27	0	0	0	0	A	0	0	0	0	0	0	0	A	0	0	0
4	12	20	28	0	0	0	A	0	0	0	0	0	0	0	A	0	0	0	0
5	13	21	29	0	0	A	0	0	0	0	0	0	0	A	0	0	0	0	0
6	14	22	30	0	A	0	0	0	0	0	0	0	A	0	0	0	0	0	0
7	15	23	31	A	0	0	0	0	0	0	0	A	0	0	0	0	0	0	0

A) 0=NO ACKNOWLEDGE / 1=ACKNOWLEDGE

Table 5-7. Multiple Slave Acknowledge Symbols (Four Words)

Data Check Line Formats (Used only for Class EC)

MODULE ID ASSIGNMENT				DATA CHECK LINES							
AWM1	AWM2	AWM3	AWM4	7	6	5	4	3	2	1	0
0	8	16	24	0	0	0	0	0	0	0	A
1	9	17	25	0	0	0	0	0	0	A	0
2	10	18	26	0	0	0	0	0	A	0	0
3	11	19	27	0	0	0	0	A	0	0	0
4	12	20	28	0	0	0	A	0	0	0	0
5	13	21	29	0	0	A	0	0	0	0	0
6	14	22	30	0	A	0	0	0	0	0	0
7	15	23	31	A	0	0	0	0	0	0	0

A) 0=NO ACKNOWLEDGE / 1=ACKNOWLEDGE

5.3 DETAILED REQUIREMENTS.

5.3.1 INTRODUCTION

Detailed requirements for the PI-bus Data Link protocol are specified in this section. The bus states which govern the cycle-by-cycle operation of the bus are defined and their relationships to the protocol states are given. The bus mastership protocol is specified, including requirements for the use of Bus Request. All PI-bus communications sequences are specified by sequence diagrams which show the scheduled sequence of bus states and corresponding module operations. Any sequence not specified herein shall be considered invalid.

The protocol for using Wait to insert non-transfer cycles into a message sequence is specified. The Data Link facilities which are required to be accessible over the bus are defined. Finally, bus response to error conditions is specified and bus diagnostics techniques are defined.

5.3.2 BUS STATE DEFINITIONS.

The protocol states defined in "5.2 GENERAL REQUIREMENTS." consist of sequences of bus states. The bus states shall define the information content of each bus cycle. Table 5-8 lists the bus states along with their corresponding Cycle Types and the protocol states in which they appear.

Table 5-8. Bus State Definitions

BUS STATES	CT	PROTOCOL STATE	COMMENT
I	I	Idle	Bus Idle cycle.
V0 .. V3	V	Vie	Vie priority bits resolved, 3 per step.
VZ0 .. VZ3	V	Vie	Non-Transfer cycle for vie decision time.
H0	H0	Header	First cycle of header transfer.
H1 .. H9	H	Header	Additional cycles of header transfer.
HZ	H	Header	Non-Transfer cycle for decision time.
HA0	A	Header Ack	Single slave or first multicast Header Acknowledge.
HA1 .. HA4	A	Header Ack	Additional multicast header acknowledge.
HAZ	A	Header Ack	Non-Transfer cycle for decision time.
D	D	Data	Datum transfer cycles.
DZ	D	Data	Non-Transfer cycle for decision time.
DA0	A	Data Ack	Single slave or first multicast Data Acknowledge.
DA1 .. DA4	A	Data Ack	Additional multicast acknowledge cycles.
S0 .. S2	S	Data	Cycles announcing message being suspended.
AB0 .. AB3	AB	Abort	Cycles announcing sequence being aborted.

Each of the PI-bus communication sequences defined herein has a corresponding sequence diagram which shows the required schedule of bus states and the corresponding bus operations for that sequence.

The Sequence diagrams contain the following information:

BUS STATE

Bus State	Unique bus state is shown for each scheduled bus cycle.

DATA

D<31..16>	Data format type or state for the most significant 16 bits of a 32 bit bus.
D<15..0>	Data format type or state for the 16 bit bus or the least significant 16 bits of a 32 bit bus.
Source	Bus Interface driving the data lines; master (M) or slave (S).
Read Source	Shows cycles where the slave (S) sources the Data during a read sequence.
Write Source	Shows cycles where the master (M) sources Data during a write sequence. If no Source, Read Source or Write Resource is shown, data lines are released.

CYCLE TYPE

Cycle Type	Defines Cycle Type symbol for each scheduled bus cycle.
Source	Shows the source of the Cycle Type symbol as the master (M) or a contander (C).

ACKNOWLEDGE SET

Acknowledge	Defines the state of the Acknowledge Set lines for each bus cycle. The AS lines may be driven by a single slave, by multiple slaves or by contenders in Vie. For the multiple slave case, an NS (Not Selected) beneath ACK means that an NS symbol may appear for that bus cycle if there are no sources for ACK during that cycle out of the group of potential slave (S) sources.

Source The Acknowledge Source is shown as slave(s) (S) or contender(s) (C) or, if not shown, the lines are released.

Source <7-0> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 0 thru 7.

Source <15-8> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 8 thru 15.

Source <23-16> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 16 thru 23.

Source <31-24> For multiple slave case, the Source maybe any slave(s) (S) with an MID value of 24 thru 31.

WAIT

Allowed Defines bus cycles where Wait may be asserted. A source for Wait is not shown in these sequences since the scheduled sequence of bus states assumes that Wait is not asserted.

A set of four colons (::::) in a sequence diagram indicates that a number of bus states occur in the sequence at that point.

5.3.3 BUS MASTERSHIP.

The protocol governing bus mastership is specified herein. The Vie and Tenure Pass Message sequences which assign bus mastership to a particular module are defined. The protocol which allows a module with higher priority than the current bus master to request a Vie sequence by asserting Bus Request is specified.

5.3.3.1 Vie Sequence.

The Vie sequence shall be used to determine a single bus master for the first Header sequence which occurs after the bus enters the Idle state. The bus master shall be selected on the basis of the Vie Priority code stored in each contender's Vie Priority Register (see "5.3.7.3.6 Vie Priority Register - Address 5."). The selected bus master may retain tenure or may assign tenure to another module by using the Tenure Pass Message sequence defined in the next section.

Any module that requires bus mastership may initiate Vie after two or more cycles of Idle. Modules shall be capable of participating in a Vie sequence which begins on the third or any later cycle of Idle. All active modules shall monitor each step of the Vie process and store the Vie Priority level of the winning module.

Due to pipeline delays, a module may attempt to initiate Vie up to one cycle after Vie is initiated by another module. In that case, the module which attempted to initiate the late Vie sequence shall cease to contend on next cycle and shall complete the original Vie sequence as a non-contender. Modules shall not attempt to initiate Vie more than one cycle after the V cycle type has been posted on the Cycle Type lines.

The Vie sequence shall be a four step sequence as defined in Table 5-9. Each vie step shall use two bus cycles to resolve three of the twelve bits of Vie Priority code. Modules which are contending for bus mastership shall decode the three most significant bits (VP<11..9>) of their Vie Priority code into a one-of-eight Module Vie Code as shown in Table 5-10 and Table 5-11. Modules shall initiate Vie by posting the Module Vie Code on the Data line group and the V cycle type on the Cycle Type lines. On the following bus cycle, contenders shall release the Data line group.

Table 5-9. Vie Sequence

	BUS STATE							
SIGNAL LINES	V0	VZ0	V1	VZ1	V2	VZ2	V3	VZ3
DATA * D<31..16> D<15..0> Source=	0 VC0 C	0 0	0 VC1 C	0 0	0 VC2 C	0 0	0 VC3 C	0 C
CYCLE TYPE Source=C	V	V	V	V	V	V	V	V
ACKNOWLEDGE Source=	NS	NS	RCG C	RCG C	RCG C	RCG C	RCG C	RCG C
WAIT Allowed	0 NO	0 NO	0 NO	0 NO	0 NO	0 NO	0 NO	0 NO
STEP SUB-CYCLE	1	2	1	2	1	2	1	2
VIE PRIORITY CODE BITS	11,10,9		8,7,6		5,4,3		2,1,0	
VIE STEP	0		1		2		3	

* - VCx is the vie code formed by the inclusive 'OR' of the vie codes asserted by all contending modules (C).

Modules shall read the logical-OR of the Module Vie Codes posted by the contenders from the Data line group at the end of the first cycle of Vie (bus state V0) as an Aggregate Vie Code (VC0). During the second cycle of each step, each contender (C) shall compare the posted Module Vie Code to the Aggregate Vie Code read from the bus. If the Aggregate Vie Code read from the bus has a bit asserted in a more significant bit position than the Module Vie Code posted by the module then the module has lost contention and shall not drive the bus on any remaining cycles in the Vie sequence. If the contender's posted Module Vie Code has the same bit asserted as the most significant bit asserted in the Aggregate Vie Code the module has not lost the vie step and shall proceed to the next Vie step as a contender. If there is an uncorrectable error in the Data line group, the module shall assume that both lines in any affected pair of redundant lines are asserted.

This process shall be repeated in the second, third and fourth vie steps using Vie Priority code bits VP<8..6>, VP<5..3> and VP<2..0>, respectively. At the conclusion of the fourth vie step, at most one module remains as a contender and that module shall become the new bus master. If a bus master is selected, uniqueness is guaranteed by the five MID bits within the Vie Priority Code. The bus master must post H0 on the bus cycle immediately following the VZ3 cycle. If no bus master is selected due to error conditions, the bus shall return to the Idle state.

Table 5-9 defines the detailed sequence of bus states and module actions during the Vie sequence. The Data lines are shown as two groups of sixteen lines each. Lines D<31..16>, if implemented, shall remain released throughout the Vie sequence. Contending modules shall post their Module Vie Codes on D<15..0> and, for Class EC buses only, on DC<7..0>. The Module Vie Codes posted by the contending modules shall be logically OR'ed during the first cycle of each vie step to form the Aggregate Vie Code for that step. Contenders shall post the symbol V on the Cycle Type lines during each cycle of Vie as illustrated. Contenders shall also post the symbol RCG on the AS lines during each cycle of the second, third and fourth vie steps. All active modules shall post NAK on the AS lines on cycle N+2 each time an uncorrectable Data group or Cycle Type group error which applies to the operation of bus cycle N is detected.

Modules shall not assert Wait nor Bus Request during the Vie sequence. Non-contenders shall not assert any line or post any symbol during the Vie sequence other than posting NAK on the AS lines as specified above. All active modules shall monitor the Vie sequence and record the winning bus master's priority code. Uncorrectable Data or Cycle Type group line errors detected during the Vie sequence shall cause the modules which do not win the Vie sequence to store "unknown" as the current bus master's priority code. During sub-cycle 1 of each Vie step, Data line errors for bit pairs other than the most significant pair that has a line asserted may be considered correctable on Class ED buses since the winning priority is not affected. A module which has "unknown" for the current bus master's priority code shall signal "command error" to the bus master if the module becomes a slave during the bus master's tenure (see "5.3.9 Error Detection, Recovery And Diagnostics.").

Table 5-10. Module Vie Code Format - Data Lines

BIT PATTERN	DATA LINES															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0 0 0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0 0 1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0
0 1 1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
1 0 0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
1 0 1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
1 1 0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
11 10 9	Vie Priority register bits for first vie step.															
8 7 6	Vie Priority register bits for second vie step.															
5 4 3	Vie Priority register bits for third vie step.															
2 1 0	Vie Priority register bits for fourth vie step.															

Table 5-11. Module Vie Code Format - Data Check Lines

BIT PATTERN	DATA CHECK LINES 7	6	5	4	3	2	1	0
0 0 0	0	0	0	0	0	0	0	1
0 0 1	0	0	0	0	0	0	1	0
0 1 0	0	0	0	0	0	1	0	0
0 1 1	0	0	0	0	1	0	0	0
1 0 0	0	0	0	1	0	0	0	0
1 0 1	0	0	1	0	0	0	0	0
1 1 0	0	1	0	0	0	0	0	0
1 1 1	1	0	0	0	0	0	0	0
11 10 9	Vie Priority register bits for first vie step.							
8 7 6	Vie Priority register bits for second vie step.							
5 4 3	Vie Priority register bits for third vie step.							
2 1 0	Vie Priority register bits for fourth vie step.							

5.3.3.2 Tenure Pass Message

The current bus master may use the Tenure Pass Message to assign bus mastership directly to another module. The current bus master may also use the Tenure Pass Message to begin a new tenure under the current or a revised priority code. However, the Tenure Pass Message shall not be performed if Bus Request is in the asserted state two cycles prior to the H0 cycle on which the Tenure Pass Message sequence would have started.

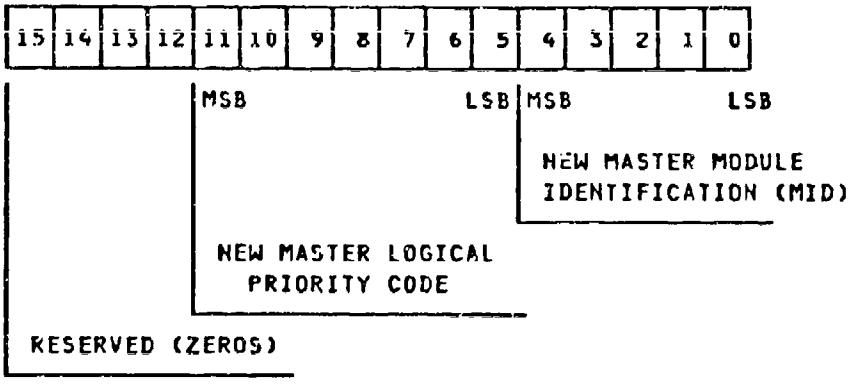
The formats which shall be used for the Tenure Pass Message header words are shown in Figure 5-6. For the Tenure Pass Message, the slave ID field of Header Word A (HWA) shall be set to the Module Identification (MID) of the module which shall become the new Bus Master. The five least significant bits of HWB must be the same as the five least significant bits of HWA. Bits five through seven of HWA must be zero. The Format (F) bit in HWA shall be 0 and the Type 16 sequence shall be used for both Type 16 and Type 32 buses. Header Word B (HWB) shall contain the new bus master priority code.

Figure 5-6. Tenure Pass Message Header Word Formats

HEADER WORD A (HWA)

AT			MSG TYPE				F	SLAVE ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	1	0	0	0	0	0	MID				

HEADER WORD B (HWB)



The scheduled sequence of bus states for the Tenure Pass Message shall be as shown in Table 5-12. Each bus state shall use one bus cycle except that

the master or slave may insert non-transfer cycles into the sequence by asserting Wait during the HZ and/or HA0 states.

To initiate a Tenure Pass Message, the bus master (M) shall post HWA on D<15..0> and post the symbol H0 on the Cycle Type (CT) lines during the H0 state of the Tenure Pass Message. The bus master shall post HWB on D<15..0> and the symbol H on the CT lines for the H1 state. During the HZ state, the Data group lines shall be released and the bus master shall post H on the CT lines. Also, the slave (S) shall post the symbol RCG on the Acknowledge Set group lines. On the HA0 state, the slave shall post the single slave Acknowledge word (AWS) on D<15..0> and shall post the symbol ACK on the AS group lines. Also, the master shall post the symbol A on the CT group lines. The bus master shall post A on the CT lines and the slave shall post ACK on the AS group lines during the HAZ state. The Data group lines shall be released during HAZ.

The original bus master shall not post Abort nor post Wait on HAZ and shall release all signal lines on the bus cycle following HAZ. The original bus master's tenure shall end with the HAZ state and the original slave's tenure as the new bus master shall begin on the next bus cycle.

All active modules shall monitor the Tenure Pass Message and shall acquire the new bus master's priority code from HWB. Any module that detects an error in the Tenure Pass Message shall store "unknown" as the current bus master's priority code. A module which has "unknown" for the current bus master's priority code shall signal "command error" to the bus master if the module becomes a slave during the bus master's tenure (see "5.3.9 Error Detection, Recovery And Diagnostics.").

The Tenure Pass Message shall not affect the Vie Priority Register of any module.

Table 5-12. Tenure Pass Message Sequence

	BUS STATE				
SIGNAL LINES	H0	H1	HZ	HA0	HAZ
DATA D<31..16> D<15..0> Source =	0 HWA M	0 HWB M	0 0	0 AWS S	0 0
CYCLE TYPE Source =M	H0	H	H	A	A
ACKNOWLEDGE Source =	NS	NS	RCG S	ACK S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 (1)

(1) Note: Only next master can assert wait on this cycle.

5.3.3.3 Bus Request.

The Bus Request line shall be asserted by a module to signal the current bus master that the module has a higher priority requirement for bus mastership. The module asserting Bus Request shall ensure that this condition is met by only asserting Bus Request when the module's Vie Priority register contains a higher priority code than that of the current bus master. A module shall not assert Bus Request when the current bus master's priority code is unknown. The current bus master shall honor Bus Request by relinquishing tenure and releasing all bus signal lines by the end of the Vie Interval defined by the sum of the bus cycles specified by the contents of the Vie Interval A Register plus the contents of the Vie Interval B Register plus six cycles (see "5.3.7.3.4 Vie Interval A Register - Address 3." and "5.3.7.3.5 Vie Interval B Register - Address 4.").

The assertion of Bus Request shall be allowed on any bus cycle, except for the cycles starting with the third cycle of Idle and continuing through the last cycle of Vie. If a module asserts Bus Request during a Tenure Pass Message and the new bus master acquires tenure with a higher priority than the module asserting Bus Request, the module shall release Bus Request within six cycles after the start of the next bus master's tenure.

The bus master shall count all bus cycles, including non-transfer cycles, whenever Bus Request is asserted. The first cycle counted shall be the first cycle after the cycle containing the initial assertion of Bus Request. Any subsequent cycle in which Bus Request is released shall not be counted and shall cause the accumulated count to be discarded. To relinquish tenure, the bus master may:

1. release all bus lines to place the bus in the Idle state rather than post an H0 cycle or

2. if a single slave Block Message data sequence is in progress and the AWT field in the associated Single Slave Header Acknowledge Word is 010 or 011, the bus master may perform a Suspend Sequence (see "5.3.5.1 Suspend.") and release all bus lines before the total Vie Interval time elapses.

If the cycle count reaches the sum of the value specified in the Vie Interval A Register plus the value specified in the Vie Interval B Register, the bus master shall perform an Abort sequence such that the first cycle of the Abort sequence occurs on the second bus cycle immediately following the cycle that exceeds the Vie Interval limit. After completing the Abort sequence the master shall immediately relinquish tenure and release all bus lines (see "5.3.5.2 Abort.").

5.3.4 MESSAGE SEQUENCES

5.3.4.1 Parameter Write Message.

The Parameter Write Message shall be used to transfer a one word parameter from the master device to a slave or multiple slave devices. The Parameter Write sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-13

Type 16, multiple slave -- Table 5-14

Type 32, single slave ---- Table 5-15

Type 32, multiple slave -- Table 5-16

Header word formats for the Parameter Write message shall be as shown in Figure 5-7. Header Word A shall specify the slave(s), Type 16 or 32 Format, Access and Message Types. The Access Type field is application dependent, except that the code 111 shall be reserved. Message Types shall be as defined for the single slave (SS) and multiple slave (MS) cases. Header Word B shall contain the parameter information to be passed to the device. Header Word C0 and C1 shall contain 32 bits of virtual addressing information to be passed to the device.

The Header Acknowledge Word (AWS) shall supply current message status information to the master from the slave for single slave sequences. For multiple slave sequences, the Multiple Slave Status Word (AWM0) shall supply current message status information to the master. The multiple slave Header Acknowledge Words (AWM1, AWM2, AWM3, AWM4) convey the list of slave participants to the master.

Figure 5-7. Parameter Write Header Word Formats

HEADER WORD A (HWA)

AT			MSG TYPE				F	SLAVE ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS TYPE			0001-SS 0011-MS												

HEADER WORD B (HWB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB <---------- PARAMETER ----------> LSB

HEADER WORD C0 (HWC0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

15 <----- LEAST SIGNIFICANT ADDRESS BITS------> 0

HEADER WORD C1 (HWC1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31 <-----MOST SIGNIFICANT ADDRESS BITS-------> 16

Table 5-13. Parameter Write Sequence - Type 16 Single Slave

	BUS STATE					
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES

Table 5-14. Parameter Write Sequence - Type 16 Multiple Slave

	BUS STATE									
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 0 HWA M	 0 HWB M	 0 HWC0 M	 0 HWC1 M	 0 0 	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS 	NS 	RCG S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-15. Parameter Write Sequence - Type 32 Single Slave

	BUS STATE			
SIGNAL LINES	H0	H1	HZ	HA0
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 0 0	 0 AWS S
CYCLE TYPE Source=M	H0	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES

Table 5-16. Parameter Write Sequence - Type 32 Multiple Slave

	BUS STATE							
SIGNAL LINES	H0	H1	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS	NS	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.4.2 Block Message - Short Header Sequence.

The Block Message - Short Header (SH) Sequence shall be used to read data from a single slave device to the master device or to write data from the master device to one or more slave devices. The Block Message - Short Header sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-17

Type 16, multiple slave -- Table 5-18

Type 32, single slave ---- Table 5-19

Type 32, multiple slave -- Table 5-20

The header word formats for the Block Message - Short Header sequence shall be as shown in Figure 5-8. Header Word A shall specify the slave(s) ID, Type 16 or 32 Format, Access and Message Types. Access Type field bits <15,14> are application dependent. Bit 13 indicates whether the message is being used to resume a previously suspended Block Message (bit 13 = 1) or send a new message (bit 13 = 0). The Message Types shall be as defined for the single slave write (SSW), single slave read (SSR) and multiple slave (MS) cases. Header Word B shall contain an unsigned binary datum count which specifies the number of datum units to be transferred between the master and slave(s), except that all zeros shall represent 65,536 datum units. The datum count shall be the number of 16 bit words for 16 bit transfers or the number of double words for 32 bit transfers. Header Word C0 and C1 shall contain 32 bits of virtual addressing information to be passed to the device. When the Block Message - Short Header is used for resuming a suspended single slave message, Header Words C0 and C1 shall be the two Resume Control Words sent from the slave to the master during the suspend sequence and they shall be returned in the order received. When the Block Message is used for resuming a suspended multicast message, the contents of C0 and C1 should be defined by application dependent convention.

The Header and Data Acknowledge Words shall supply current message status information to the master from the slave.

Data word formats are application specific. The number of words or double words transferred for each Block Message - Short Header sequence shall be equal to the value in Header Word B. For the Type 32 sequences, data words are shown with L or H designations. The least significant half of a double word or a single word transmitted on data lines D<15..0> contains the L designation. The most significant half of a double word or a single word transmitted on data lines D<31..16> contains the H designation.

Figure 5-8. Block Message - Short Header Word Formats

HEADER WORD A (HWA)

AT			MSG TYPE				F	SLAVE ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS TYPE			0101-SSW 0100-SSR 0111-MS												

HEADER WORD B (HWB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB <---------- DATUM COUNT ----------> LSB

HEADER WORD C0 (HWC0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

15 <----- LEAST SIGNIFICANT ADDRESS BITS------> 0

HEADER WORD C1 (HWC1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31 <-----MOST SIGNIFICANT ADDRESS BITS-------> 16

Table 5-17. Block Message - SH Sequence - Type 16 Single Slave

	BUS STATE										
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0	D0	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 0	0 AWS S	0 D0 S M	:::: :::: S M	0 Dn S M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	A	D	::::	D	D	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	ACK S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-18. Block Message - SH Sequence - Type 16 Multiple Slave

	BUS STATE									
SIGNAL LINES	H0	H1	H2	H3	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 0 HWA M	 0 HWB M	 0 HWC0 M	 0 HWC1 M	 0 0 	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS 	NS 	RCG S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-18. Block Message - SH Sequence - Type 16 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 0 D0 M	 :::: :::: M	 0 Dn M	 1 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-19. Block Message - SH Sequence - Type 32 Single Slave

	BUS STATE								
SIGNAL LINES	H0	H1	HZ	HA0	D0	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	HWB HWA M	HWC1 HWC0 M	0 0	0 AWS S	D0H D0L S M	:::: :::: S M	DnH DnL S M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	A	D	::::	D	D	A
ACKNOWLEDGE Source=	NS	NS	RCG S	ACK S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-20. Block Message - SH Sequence - Type 32 Multiple Slave

	BUS STATE							
SIGNAL LINES	H0	H1	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS 	NS 	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-20. Block Message - SH Sequence - Type 32 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 D0H D0L M	 :::: :::: M	 DnH DnL M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.4.3 Block Message - Extended Header Sequence.

The Block Message - Extended Header (EH) Sequence shall be used to read data from a single slave device to the master device or to write data from the master device to one or more slave devices. This sequence shall be used where more header information is required than that provided by the Block Message short header sequence. The Block Message - Extended Header sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-21

Type 16, multiple slave -- Table 5-22

Type 32, single slave ---- Table 5-23

Type 32, multiple slave -- Table 5-24

Header word formats for the Block Message - Extended Header sequence shall be as shown in Figure 5-9. Header Word A shall specify the slave(s) ID, Type 16 or 32 Format, Access and Message Types. Access Type field bits <15,14> are application dependent. Bit 13 indicates whether the message is being used to resume a previously suspended Block Message (bit 13 = 1) or send a new message (bit 13 = 0). The Message Types shall be as defined for the single slave write (SSW), single slave read (SSR) and multiple slave (MS) cases. Header Word B shall contain an unsigned binary datum count which specifies the number of datum units to be transferred between the master and slave(s), except that all zeros shall represent 65,536 datum units. The datum count shall be the number of 16 bit words for 16 bit transfers or the number of double words for 32 bit transfers. Header Word C0 and C1 shall contain 32 bits of virtual addressing information to be passed to the device. When the Block Message - Extended Header is used for resuming a suspended single slave message, Header Words C0 and C1 shall be the first two Resume Control words sent from the slave to the master during the suspend sequence and shall be returned in the order received. Header Words D0 thru D5 are also application dependent and shall be passed to the slave device. When resuming a suspended single slave message these words shall be the last six Resume Control Words sent from the slave during the suspend sequence and shall be returned in the order received. When the Block Message is used for resuming a suspended multiple slave message, the contents of HWC0, HWC1 and HWD0 through HWD5 should be defined by application dependent convention.

Figure 5-9. Block Message - Extended Header Word Formats

HEADER WORD A (HWA)

AT			MSG TYPE				F	SLAVE ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS TYPE			1101-SSW 1100-SSR 1111-MS												

HEADER WORD B (HWB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB <---------- DATUM COUNT ----------> LSB

HEADER WORD C0 (HWC0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

15 <----- LEAST SIGNIFICANT ADDRESS BITS------> 0

HEADER WORD C1 (HWC1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

31 <-----MOST SIGNIFICANT ADDRESS BITS-------> 16

HEADER WORD D0 (HWD0)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB LSB

* * *
* * *
* * *

HEADER WORD D5 (HWD5)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB LSB

Table 5-21. Block Message - EH Sequence - Type 16 Single Slave

	BUS STATE								
SIGNAL LINES	H0	H1	H2	H3	H4	::::	H9	HZ	HA0
DATA D<31..16> D<15..0> Source=	0 HWA M	0 HWB M	0 HWC0 M	0 HWC1 M	0 HWD0 M	:::: :::: M	0 HWD5 M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	::::	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-21. Block Message - EH Sequence - Type 16 Single Slave (continued)

	BUS STATE						
SIGNAL LINES	D0	D1	D2	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 0 D0 S M	 0 D1 S M	 0 D2 S M	 :::: :::: S M	 0 Dn S M	 0 0 	 0 AWS S
CYCLE TYPE Source=M	D	D	D	::::	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-22. Block Message - EH Sequence - Type 16 Multiple Slave

	BUS STATE										
SIGNAL LINES	H0	H1	H2	H3	H4	::::	H9	HZ	HA0	HA1	HA2
DATA D<31..16> D<15..0> Source=	 0 HWA M	 0 HWB M	 0 HWC0 M	 0 HWC1 M	 0 HWD0 M	 :::: :::: M	 0 HWD5 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S
CYCLE TYPE Source=M	H0	H	H	H	H	::::	H	H	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS	NS	RCG S S S S	RCG S S S S	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-22. Block Message - EH Sequence - Type 16 Multiple Slave (continued)

	BUS STATE										
SIGNAL LINES	HA3	HA4	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 0 AWM3 S	 0 AWM4 S	 0 D0 M	 :::: :::: M	 0 Dn M	 0 0 	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	A	A	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	ACK (NS) S 	ACK (NS) S	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-23. Block Message - EH Sequence - Type 32 Single Slave

	BUS STATE						
SIGNAL LINES	H0	H1	H2	H3	H4	HZ	HA0
DATA D<31..16> D<15..0> Source=	HWB HWA M	HWC1 HWC0 M	HWD1 HWD0 M	HWD3 HWD2 M	HWD5 HWD4 M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	H	H	H	A
ACKNOWLEDGE Source=	NS	NS	RCG S	RCG S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-23. Block Message - EH Sequence - Type 32 Single Slave (continued)

	BUS STATE						
SIGNAL LINES	D0	D1	D2	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 D0H D0L S M	 D1H D1L S M	 D2H D2L S M	 :::: :::: S M	 DnH DnL S M	 0 0 	 0 AWS S
CYCLE TYPE Source=M	D	D	D	::::	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-24. Block Message - EH Sequence - Type 32 Multiple Slave

	BUS STATE										
SIGNAL LINES	H0	H1	H2	H3	H4	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 HWB HWA M	 HWC1 HWC0 M	 HWD1 HWD0 M	 HWD3 HWD2 M	 HWD5 HWD4 M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	H	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS 	NS 	RCG S S S S	RCG S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-24. Block Message - EH Sequence - Type 32 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 D0H D0L M	 :::: :::: M	 DnH DnL M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	RCG S S S S	:::: S S S S	RCG S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.4.4 Bus Interface Message Sequence

The Bus Interface Message shall be used to read or write to the Bus Interface register address spaces. The Bus Interface Message sequence of bus states shall be as defined in the following tables:

Type 16, single slave ---- Table 5-25

Type 16, multiple slave -- Table 5-26

Header word formats for the Bus Interface Message sequence shall be as shown in Figure 5-10. Header Word A shall specify the participant slave(s), Type 16 Format, Access and Message Types. The same format shall be used for both Type 16 and Type 32 buses. Access Type code 000 is defined for the Data Link register address space. Codes 001 through 011 shall be reserved for higher level protocols and the code 111 shall be reserved for future use. The other codes may be used for implementation defined registers. The Message Types shall be as defined for the single slave write (SSW), single slave read (SSR) and multiple slave (MS) cases. Header Word B shall contain the address for the first Bus Interface register to be accessed in the sequence and an unsigned binary word count specifying the number of data words to be transferred between the master and the slave(s), except that a word count of all zeros shall mean 256 words. Each data transfer after the first data transfer shall access a successive register address. The register address shall be incremented after each data word transfer.

Data word formats are as defined in "5.3.7 Data Link Facilities.." The number of data words transferred shall equal the value in Header Word B.

Figure 5-10. Bus Interface Message Header Word Formats

HEADER WORD A (HWA)

AT			MSG TYPE				F	SLAVE ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS TYPE			1001-SSW 1000-SSR 1011-MS				0								

HEADER WORD B (HWB)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSB							LSB	MSB							LSB
REGISTER ADDRESS								WORD COUNT							

Table 5-25. Bus Interface Message Sequence - Type 16 Single Slave

	BUS STATE								
SIGNAL LINES	H0	H1	HZ	HA0	D0	::::	Dn	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	0 HWA M	0 HWB M	0 0	0 AWS S	0 D0 S M	:::: :::: S M	0 Dn S M	0 0	0 AWS S
CYCLE TYPE Source=M	H0	H	H	A	D	::::	D	D	A
ACKNOWLEDGE Source=	NS	NS	RCG S	ACK S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-26. Bus Interface Message Sequence - Type 16 Multiple Slave

	BUS STATE							
SIGNAL LINES	H0	H1	HZ	HA0	HA1	HA2	HA3	HA4
DATA D<31..16> D<15..0> Source=	 0 HWA M	 0 HWB M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	H0	H	H	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	NS	NS	RCG S S S S	ACK S S S S	ACK (NS) S	ACK (NS) S	ACK (NS) S	ACK (NS) S
WAIT Allowed	0 NO	0 NO	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-26. Bus Interface Message Sequence - Type 16 Multiple Slave (continued)

	BUS STATE								
SIGNAL LINES	D0	::::	Dn	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 0 D0 M	 :::: :::: M	 0 Dn M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	D	::::	D	D	A	A	A	A	A
ACKNOWLEDGE	RCG	::::	RCG	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT Allowed	0 YES	:::: YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.5 EXCEPTION SEQUENCES.

5.3.5.1 Suspend.

The current bus master may elect to suspend a Block Message during the Data sequence providing that there are at least three Data Cycles remaining and the corresponding Header Acknowledge Word had an S of 1 and, for single slave messages, an AWT of 01 or 10. After completing the Suspend sequence, the current bus master may initiate a new message with an H0 cycle or release the bus to initiate the Idle state.

The Suspend Sequence may be used in conjunction with the Vie Interval A and Vie Interval B Registers to optimize the trade-off between short bus acquisition latency and large message sizes by allowing a Block Message to be interrupted and resumed at a later time.

5.3.5.1.1 Single Slave Suspend.

The Suspend sequence of bus states for single slave Block Messages shall be as defined in the following tables:

Type 16, single slave, short data ---- Table 5-27

Type 32, single slave, short data ---- Table 5-28

Type 16, single slave, extended data - Table 5-29

Type 32, single slave, extended data - Table 5-30

The Short Data Sequence shall be used if the AWT field of the Single Slave Acknowledge Word from the suspended message was 01. The Extended Data Sequence shall be used if the AWT field was 10. In terms of the data transfer operation, the S cycles of these sequences shall be considered normal D cycles and all Bus Interfaces shall respond accordingly. The slave shall post ACK during the last of the three S cycles to indicate that the Suspend Sequence has been recognized.

Beginning on the fourth cycle of a Suspend sequence, the slave shall transmit to the bus master implementation/device specific Resume Control Words that the master shall store for later use in resuming the message. The Suspend - Short Data sequences transfer two Resume Control Words and the Suspend - Extended Data sequences transfer eight Resume Control Words. The Resume Control Words shall be followed by a Data Acknowledge Word.

5.3.5.1.2 Multiple Slave Suspend.

The Suspend sequence of bus states for multiple slave Block Messages shall be as defined in the following tables:

Type 16, multiple slave ---- Table 5-31

Type 32, multiple slave ---- Table 5-32

In terms of the data transfer operation, the S cycles of these sequences shall be considered normal D cycles and all Bus Interfaces shall respond accordingly. The slaves shall post ACK during the last of the three S cycles to indicate that the Suspend Sequence has been recognized. A non-transfer DZ cycle and a multiple slave Data Acknowledge sequence shall follow the third S cycle. No Resume Control Words are sent from the slaves to the master.

5.3.5.1.3 Resuming Suspended Messages

A master shall resume a suspended Block Message by transmitting the remaining data to the slave(s) in a Block Message with bit 13 of the HWA AT field set to 1. The Resume Control Words that the slave(s) may require to resume that message shall be sent to the slave(s) during the appropriate header cycles as described in "5.3.4.2 Block Message - Short Header Sequence." and "5.3.4.3 Block Message - Extended Header Sequence.." For a single slave message, the Resume Control Words shall be those words transmitted from the slave to the master during the suspend sequence. For a multiple slave message, the Resume Control Words must be created by the master according to the slaves' resume control requirements.

A Block Message - Short Header shall be used to resume a suspended single slave Block Message which had an AWT field of 01 (two Resume Control Words) and a Block Message - Extended Header shall be used to resume a suspended single slave Block Message which had an AWT field of 10 (eight Resume Control Words). The type of Block Message used to resume a multiple slave message shall be determined by the slaves' resume control word requirements.

Note that the choice of a short or extended header Block Message for the resume sequence is not determined by whether a short or extended header was used in the suspended Block Message.

5.3.5.1.4 Using Suspend With Vie Intervals A and B

The PI-bus Vie Interval A and Vie Interval B Registers may be used by some bus interfaces to determine when a Suspend sequence is required to meet the Vie Interval requirement and to limit the number of non-transfer cycles caused by assertion of Wait during a Suspend sequence. Vie Interval A may be selected to define a checkpoint where the master bus interface determines whether or not to suspend an incomplete Block Message. Vie Interval B may be selected to define the number of bus cycles allowed for completion of the Suspend sequence, including non-transfer cycles that may be required as a result of Waits. If the actual number of Waits causes the total bus cycles to exceed the number specified by the contents of the Vie Interval A Register plus the contents of the Vie Interval B Register, the master shall perform an Abort sequence such that the first cycle of the Abort sequence occurs on the second bus cycle immediately following the cycle that exceeds the Vie Interval limit. This Abort cycle shall be the first cycle in the Abort Sequence, described in "5.3.5.2 Abort.." At the end of the Abort sequence the master shall immediately release the bus to the Idle state.

Table 5-27. Suspend - Short Data Sequence - Type 16 Single Slave

	BUS STATE						
SIGNAL LINES	Di	Di+1	Di+2	RD0	RD1	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 0 Di S M	 0 Di+1 S M	 0 Di+2 S M	 0 RCD0 S	 0 RCD1 S	 0 0	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-28. Suspend - Short Data Sequence - Type 32 Single Slave

	BUS STATE					
SIGNAL LINES	Di	Di+1	Di+2	RD0	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 DiH DiL S M	 Di+1H Di+1L S M	 Di+2H Di+2L S M	 RCD1 RCD0 S	 0 0	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-29. Suspend - Extended Data Sequence - Type 16 Single Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	RD0	RD1	::::	RD7	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 0 Di S M	 0 Di+1 S M	 0 Di+2 S M	 0 RCD0 S	 0 RCD1 S	 :::: :::: S	 0 RCD7 S	 0 0	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	::::	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	:::: S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	:::: YES	0 YES	0 YES	0 YES

Table 5-30. Suspend - Extended Data Sequence - Type 32 Single Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	RD0	RD1	RD2	RD3	DZ	DA0
DATA D<31..16> D<15..0> Source= Read Source= Write Source=	 DiH DiL S M	 Di+1H Di+1L S M	 Di+2H Di+2L S M	 RCD1 RCD0 S	 RCD3 RCD2 S	 RCD5 RCD4 S	 RCD7 RCD6 S	 0 0	 0 AWS S
CYCLE TYPE Source=M	S	S	S	D	D	D	D	D	A
ACKNOWLEDGE Source=	RCG S	RCG S	ACK S	RCG S	RCG S	RCG S	RCG S	RCG S	ACK S
WAIT Allowed Source=	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-31. Suspend - Type 16 Multiple Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 0 Di M	 0 Di+1 M	 0 Di+2 M	 0 0 	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	S	S	S	D	A	A	A	A	A
ACKNOWLEDGE Source(7..0)= Source(15..8)= Source(23..16)= Source(31..24)=	RCG S S S S	RCG S S S S	ACK S S S S	RCG S S S S	ACK S S S S	ACK (NS) S 	ACK (NS) S 	ACK (NS) S 	ACK (NS) S
WAIT Allowed	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

Table 5-32. Suspend - Type 32 Multiple Slave

	BUS STATE								
SIGNAL LINES	Di	Di+1	Di+2	DZ	DA0	DA1	DA2	DA3	DA4
DATA D<31..16> D<15..0> Source=	 DiH DiL M	 Di+1H Di+1L M	 Di+2H Di+2L M	 0 0	 0 AWM0 S	 0 AWM1 S	 0 AWM2 S	 0 AWM3 S	 0 AWM4 S
CYCLE TYPE Source=M	S	S	S	D	A	A	A	A	A
ACKNOWLEDGE	RCG	RCG	ACK	RCG	ACK	ACK (NS)	ACK (NS)	ACK (NS)	ACK (NS)
Source(7..0)=	S	S	S	S	S	S			
Source(15..8)=	S	S	S	S	S		S		
Source(23..16)=	S	S	S	S	S			S	
Source(31..24)=	S	S	S	S	S				S
WAIT Allowed	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES	0 YES

5.3.5.2 Abort.

The Abort Sequence may be used to terminate a message prior to completion due to errors or other reasons. No header or acknowledge words are required for the Abort sequence. A master may initiate the Abort Sequence at any time during its tenure, except the last cycle of a Tenure Pass Message.

The Abort Sequence shall be as shown in Table 5-33. The first three of the four bus cycles with the AB cycle type are used to give the slave time to recognize that an abort has occurred. The slave (or slaves in the case of multicast or broadcast) shall respond by posting ACK on the AS lines during the third AB cycle to inform the master that an Abort has been recognized. There are no slaves on the fourth AB cycle and the master is the only module that may assert Wait on that cycle. After the last AB cycle, the master may continue with the HO of another message or relinquish control of the bus by releasing the bus signal lines.

Table 5-33. Abort Sequence

	BUS STATE			
SIGNAL LINES	AB0	AB1	AB2	AB3
DATA D<31..16> D<15..0> Source=	 X X M or S	 X X M or S	 0 0	 0 0
CYCLE TYPE Source=M	AB	AB	AB	AB
ACKNOWLEDGE Source=	X X	X X	ACK S	NS
WAIT Allowed Source=	0 a a	0 a a	0 NO	0 b b

a) Slave Wait assertion allowed
but not honored by master
b) Master Wait allowed,
no slaves exist
X) not defined

5.3.6 WAIT.

A master or slave may assert Wait to control the data transfer rate on the PI-bus and thereby accommodate slow or temporarily busy devices. During a multiple slave sequence, one or more of the slaves may assert Wait. Each cycle on which Wait is asserted shall cause the insertion of a non-transfer cycle into a sequence.

5.3.6.1 Rules for Asserting Wait.

A Bus Interface may assert Wait on one or more cycles subject to the following restrictions:

1. Wait may normally be asserted by a Bus Interface only if that module is scheduled to be a bus master or slave on the next transfer cycle. However, the current bus master may assert Wait if the bus will be placed in the Idle state following the Wait induced non-transfer cycle(s) provided

the vie interval requirement is not violated,

2. A Bus Interface shall not assert Wait on H0 or H1 cycles.

3. A Bus Interface shall not assert Wait on a particular cycle (N), if on the second previous cycle (N-2) the module did not assert Wait but Wait was asserted on that cycle. If the Bus Interface does assert Wait on cycle N and Wait was not asserted by any module on cycle N-1, the Bus Interface shall assert Wait for an even number of contiguous cycles.

5.3.6.2 Effects of Wait.

Other than during an Abort sequence, for each cycle that a Wait is asserted during a tenure, one non-transfer cycle (NT cycle) shall be inserted into the sequence immediately following the cycle in which Wait is asserted. The insertion of non-transfer cycles shall not change the sequence of scheduled bus states. However, if during the NT cycle, the master asserts an Abort sequence designation on the Cycle Type lines, the sequence shall be altered to Abort. During an Abort sequence, Wait shall not introduce non-transfer cycles.

5.3.6.2.1 Line Groups during Non-Transfer Cycles

During NT cycles produced by Wait, the signal line values shall be as specified below.

5.3.6.2.1.1 Data Line Group during NT Cycles. The value on the Data lines shall be ignored except for line error checking. Only valid symbols shall be posted on the Data line group.

The Data line group shall not be posted by any module other than the module which would have posted the Data lines had that cycle been the scheduled transfer cycle. If a module asserts a value on the Data line group on such an NT cycle, the value shall be a symbol which is valid for the next scheduled transfer cycle.

5.3.6.2.1.2 Cycle Type Group during NT Cycles. The value on the CT lines shall be ignored except for line error checking and the occurrence of Abort. An Abort Cycle Type shall mark the beginning of an Abort Sequence. The master module responsible for posting the CT lines on the next scheduled transfer cycle shall source a valid symbol on the CT lines during the NT cycle(s).

5.3.6.2.1.3 AS Group during NT Cycles. If a module is responsible for posting the AS lines on the next transfer cycle it shall post a valid symbol on the AS lines during the NT cycle(s). A symbol other than NAK shall be ignored during non-transfer cycles. NAK shall be posted on cycle N, to signal that an error associated with cycle N-2 was detected.

5.3.6.2.1.4 Wait lines during NT Cycles. The rules for asserting Wait are specified in "5.3.6.1 Rules for Asserting Wait.." The effects of asserting Wait on a non-transfer cycle shall be the same as specified herein for assert-

ing Wait on a transfer cycle.

5.3.6.2.1.5 Bus Request Lines during NT Cycles. The rules for asserting Bus Request are specified in "5.3.3.3 Bus Request.." Bus Request is completely independent of Wait.

5.3.7 DATA LINK FACILITIES.

This section specifies the Data Link facilities which shall be accessible over the PI-bus via the Bus Interface Message described in "5.3.4.4 Bus Interface Message Sequence."

5.3.7.1 Data Link Register Address Space.

The Data Link facilities specified herein shall be contained in the Data Link register address space which shall consist of 256 words assigned to consecutive addresses. This register space shall be allocated as shown in Table 5-34. Access to these facilities over the bus shall be allowed only via the Bus Interface Message with the Header Word A Access Type field set to zero (AT=000).

Reserved registers shall not be implemented and any attempt to access them shall cause a 'Resource Not Present' error. For write operations to defined registers, the state of any bits in the word which correspond to reserved bits shall be ignored.

Table 5-34. Data Link Register Address Space

ADDRESS	REGISTER
255 . . . 33	LOGICAL SLAVE IDENTIFICATION REGISTERS (33 - 255)
32 . . 6	RESERVED REGISTERS (6 - 32)
5	VIE PRIORITY REGISTER
4	VIE INTERVAL B REGISTER
3	VIE INTERVAL A REGISTER
2	MODULE CAPABILITIES REGISTER
1	CONTROL REGISTER
0	MULTICAST ACKNOWLEDGE REGISTER

5.3.7.2 Register Protection.

Access to the Data Link registers via the Bus Interface Message shall be limited by write protection. Either permanent or programmable write protection shall be provided for each register as specified in the following sections. The current state of programmable write protect must be controlled from the device. On reset, all programmable protection must be placed in the non-protect state.

5.3.7.3 Registers.

5.3.7.3.1 Multicast Acknowledge Register - Address 0.

The Multicast Acknowledge Register shall be a 16 bit register with the same format and field interpretations as the Single Slave Acknowledge word. During Multicast sequences, error information shall be accumulated by the slave. This information is identical to that required for the single slave case The acknowledge word which would have been transferred to the master if the current sequence was a single slave sequence, shall be stored in the Multicast Acknowledge Register instead of being sent as an AWS, and the Multicast Acknowledge symbol shall be posted on the bus. Error indications shall apply to the current message only.

During multiple slave Bus Interface Message and Block Message sequences, a word equivalent to the Data Acknowledge word for a single slave sequence shall be formed. The acknowledge information stored in Multicast Acknowledge register bits <14..7> on the Header Acknowledge cycle shall be logically OR'ed with bits <14..7> of the equivalent single slave Data Acknowledge word and the result stored in bits <14..7> of the Multicast Acknowledge register on the slave's assigned Data Acknowledge cycle. Bit <15> and bits <6..0> of the equivalent single slave Data Acknowledge word shall be stored in Multicast Acknowledge Register bit <15> and bits <6..0> on the slave's assigned Data Acknowledge cycle. This Register shall be reset at the start of a multi-slave message in which this module is a participant.

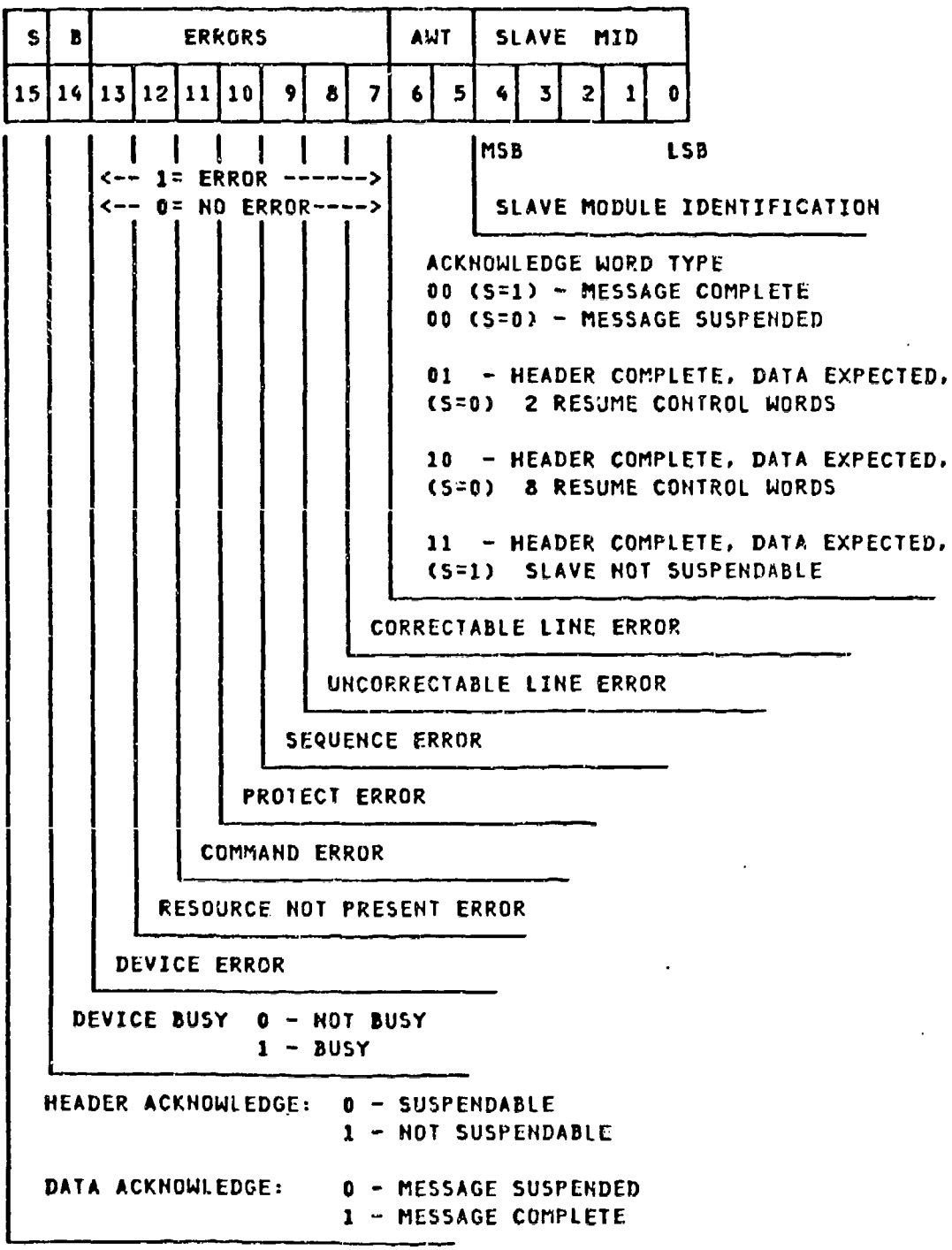
The following requirements shall apply to the Multicast Acknowledge Register:

Word format: Figure 5-11

Write protect: Permanent

State after reset: Bit <15>=1, Bits <14..5>=all zeros, Bits <4..0>=MID

Figure 5-11. Multicast Acknowledge Register Word Format



5.3.7.3.2 Control Register - Address 1.

The Control Register shall be a 2 bit register that shall contain a bit to initiate Bus Interface reset and a bit to initiate built-in-test.

The following requirements shall apply to the Control Register:

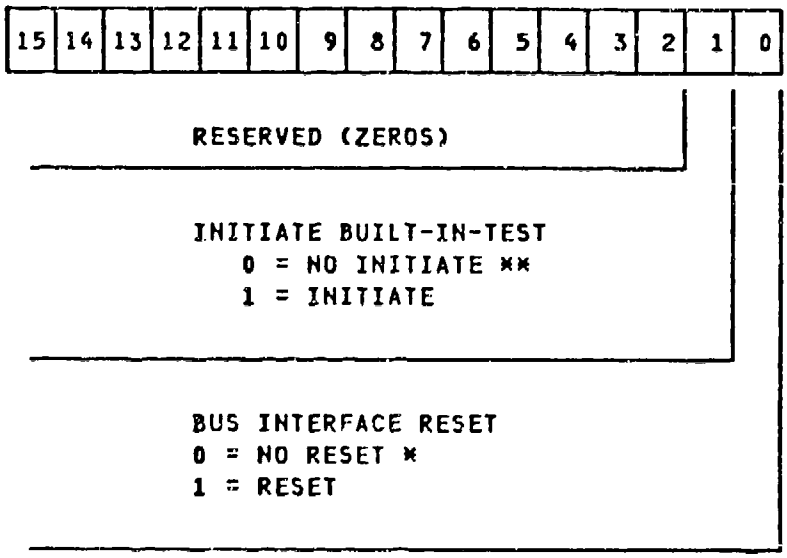
Word format: Figure 5-12

Write protect: programmable

State after reset: all zeros.

Reserved bits: read as zeros.

Figure 5-12. Control Register Word Format



* - Placed in 0 state at completion of reset.
** - Placed in 0 state at completion of 'built-in-test'.

5.3.7.3.3 Module Capabilities Register - Address 2.

The Module Capabilities Register shall be a 3 bit register that shall specify the physically implemented capabilities of the Bus Interface / Device Combination.

The following requirements shall apply to the Module Capabilities register:

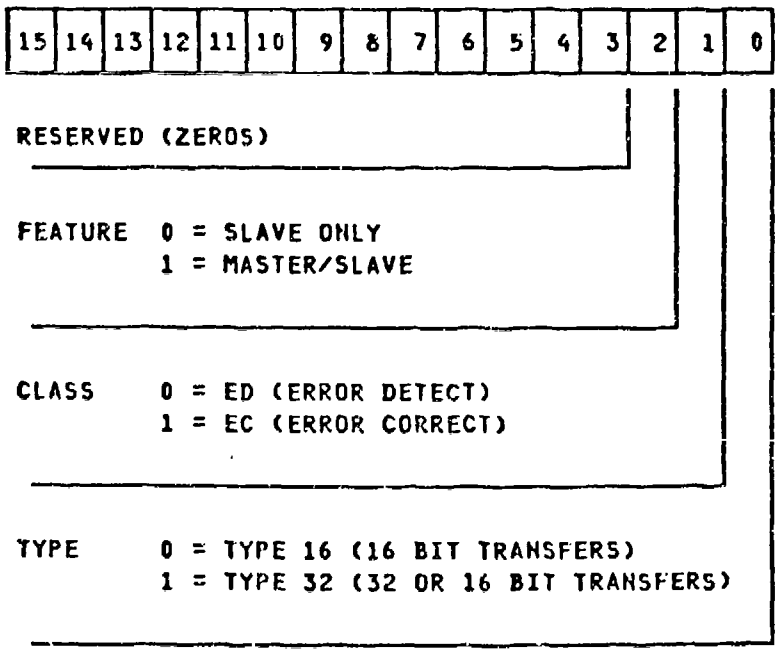
Word format: Figure 5-13

Write protect: permanent

State after reset: appropriate capabilities code

Reserved bits: read as zeros.

Figure 5-13. Module Capabilities Register Word Format



5.3.7.3.4 Vie Interval A Register - Address 3.

The Vie Interval A Register shall be a 16 bit register that shall contain the unsigned binary Vie Interval A timeout value expressed in bus cycles. The value in this register shall remain unchanged until a new timeout value is loaded. Reading this register shall return the last value written. This register is not required for Feature S0 modules. Any attempt to access a register which is not implemented shall cause a 'resource not present' error.

The following requirements shall apply to the Vie Interval A Register:

Word format: Figure 5-14

Write protect: programmable

State after reset: all ones

Figure 5-14. Vie Interval A Register Word Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB <------ VIE INTERVAL A VALUE ----------> LSB

5.3.7.3.5 Vie Interval B Register - Address 4.

The Vie Interval B Register shall be an 8 bit register that shall contain the unsigned binary Vie Interval B timeout value expressed in bus cycles. The value in this register shall remain unchanged until a new timeout value is loaded. Reading this register shall return the last value written. This register is not required for Feature S0 modules. Any attempt to access a register which in not implemented shall cause a 'resource not present' error.

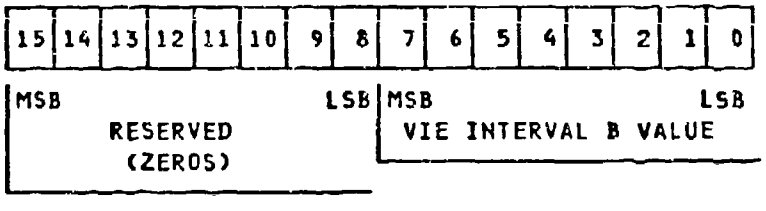
The following requirements shall apply to the Vie Interval B Register:

Word format: Figure 5-14

Write protect: programmable

State after reset: Bits <15..8>, all zeros; bits<7..0>, all ones

Figure 5-15. Vie Interval B Register Word Format



5.3.7.3.6 Vie Priority Register - Address 5.

The Vie Priority Register shall be a 12 bit register that shall contain the current vie priority of this module expressed in unsigned binary notation. The priority range is from zero (lowest priority), to 4095 (highest priority). This register is divided into two fields. The module identification field shall be fixed by the module's hardwired module identification code (MID) and cannot be changed by the Bus Interface Message. This register is not required for Feature 50 modules. Any attempt to access a register which is not implemented shall cause a 'resource not present' error.

The following requirements shall apply to the Vie Priority register:

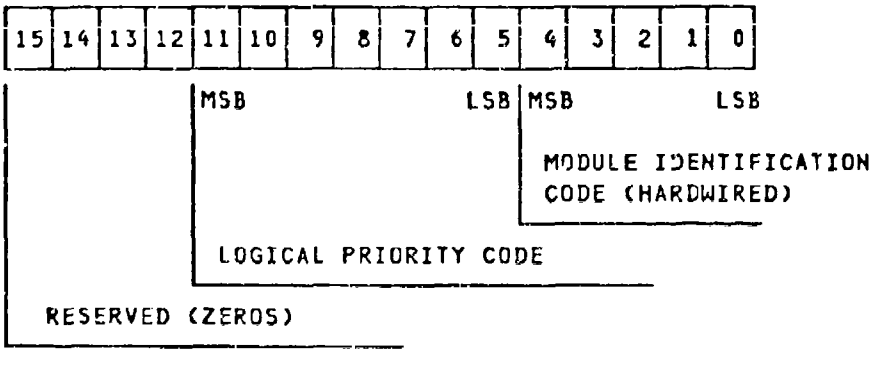
Word format: Figure 5-16

Write protect: Programmable for bits<11..5>; bits<5..0> fixed by MID

State after reset: Bits<11..5>, all zeros; bits<4..0>= MID bits<4..0>

Reserved bits: read as zeros.

Figure 5-16. Vie Priority Register Word Format



5.3.7.3.7 Reserved Registers - Addresses 6 To 32.

The use of the Reserved Registers shall be defined only by future versions of this specification. Until then, these registers shall not be implemented and shall cause a 'Resource Not Present' error if an access is attempted.

5.3.7.3.8 Logical Slave Identification Registers - Addresses 33 to 255.

The Logical Slave Identification Registers consist of a set of one bit registers in locations 33 to 255 of the Data Link register address space. The address of each register is identical to the Slave Identification code which the register controls. Bit 0 of each Logical Slave ID register indicates whether or not a Slave Identification code (ID) equivalent to the register address will be recognized by the module as a valid slave ID. The interpretation of bit 0 shall be:

1 = Bus Interface shall recognize this address as a slave ID.
0 = Bus Interface shall not recognize this address as a slave ID.

The following specifications shall apply to the Logical Slave Identification Registers:

Write protect: programmable

State after reset: zero

Reserved bits (15-1): read as zeros.

This register set is optional. A subset of the Logical Slave ID Registers may be implemented using either one or a combination of the two methods specified below:

1. The Logical Slave ID Registers may be implemented directly as a set of one bit registers in the Bus Interface register space. The slave identification registers shall have consecutively decreasing addresses starting at address 255. Any number of registers may be implemented from zero up to the full set of 223. The module shall not respond to sequences with Slave ID's corresponding to registers which are not implemented. Any attempt to access registers that are not implemented via a Bus Interface Message shall cause a 'resource not present' error.

2. The Logical Slave ID registers may be implemented indirectly using techniques such as associatively searching for valid slave ID's. In that case, the implementation shall specify the number of unique Logical Slave ID's which can be recognized (1 to 223) and each of those shall be capable of being set by the standard Bus Interface Message to any value in the range 33 to 255, inclusive. The standard Bus Interface Message shall also be capable of invalidating any previously validated slave identification code. Any attempt to validate more logical slave ID's than

allowed by the implementation shall cause a 'resource not present' error.

Figure 5-17. Logical Slave Identification Register Word Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

RESERVED (ZEROS)

0 = DO NOT RECOGNIZE THIS REGISTER ADDRESS
AS A VALID SLAVE ID

1 = RECOGNIZE THIS REGISTER ADDRESS
AS A VALID SLAVE ID

5.3.8 INITIALIZATION.

A reset mechanism shall be provided to initialize the Bus Interface. Reset shall be invoked by the attached device or by completion of a Bus Interface Message that set bit 0 (when not write protected) of the control register (see "5.3.7.3.2 Control Register - Address 1."). The Bus Interface shall respond to reset by becoming inactive, releasing all bus drivers and initializing Data Link registers to the values defined in "5.3.7 Data Link Facilities.." Following register initialization, modules may become active on the bus provided that MID<4..0>//MIP<0> satisfies P(MID<4..0>//MIP<0>) = 1. A module with incorrect MID parity, that is P(MID<4..0>//MIP<0>) = 0, shall not assert any PI-bus signal and shall not participate in PI-bus operations. A potential bus master module shall not initiate a Vie sequence prior to determining that the bus has been Idle for a minimum of two bus cycles and shall not assert Bus Request (BR) prior to determining that the current bus master is operating at a lower priority than the potential bus master's priority.

5.3.9 ERROR DETECTION, RECOVERY AND DIAGNOSTICS.

This section describes the detection of and responses to line, sequence and semantic errors. This section also defines PI-bus diagnostics requirements.

5.3.9.1 Correctable Line Errors

If the bus symbol error is correctable, then the symbol shall be functionally interpreted as the valid symbol with the least coding distance from the erroneous symbol. During message sequences, errors shall be logged in the Acknowledge word by setting the 'Correctable Line Error' bit to one (refer to "5.2.3.3.1 Single Slave Acknowledge."). The device should be notified of detected errors.

5.3.9.2 Uncorrectable Line Errors

If the bus symbol is uncorrectable, then the symbol shall be functionally interpreted according to Table 5-35. Any slave that detects an uncorrectable line error on the Data or CT lines shall post NAK on the second cycle after the cycle to which the error applies. During Vie, any module that detects an uncorrectable line error on the Data or CT lines shall post NAK on the second cycle after the cycle to which the error applies. The device should be notified of line errors. During messages, the error shall be logged in the Acknowledge word by setting the 'Uncorrectable Line Error' bit to one (refer to "5.2.3.3.1 Single Slave Acknowledge.").

Table 5-35. Interpretation and Response for Uncorrectable Invalid Symbols

SIGNAL LINE GROUP	INTERPRETATION AND RESPONSE
Wait	If Wait is not allowed: No Wait. If Wait is allowed: Wait asserted on first cycle of detected error and No Wait asserted on remaining contiguous error cycles.
Cycle Type	Scheduled Cycle Type (per defined sequences).
Cycle Type (during H0)	No slave ID match.
Bus Request	No Bus Request asserted.
Acknowledge Set	Assume NAK posted.
Acknowledge Set (during Vie)	If not the winner of vie, set master priority unknown. Should notify device.
Data lines (during Vie)	If a contender, assume that redundant bits which disagree are both asserted. If not a contender, set master priority unknown. If not the winner of vie, set master priority unknown.
Data lines (during Multi-slave acknowledgement)	For redundant bits which disagree, no acknowledgement for corresponding module.
Data lines (during H0)	No slave ID match.
Data lines (during header, except H0)	Slaves shall, discard header words, take no action based on header words, set the Acknowledge Word AWT field to 00, set the S field to 1 and become not selected after Header Acknowledge

5.3.9.3 Sequence Errors.

Each Bus Interface shall detect any error that causes a violation in the protocol syntax (sequence definitions). During message sequences, errors shall be logged in the Acknowledge word by setting the 'Sequence Error 'bit to one (refer to "5.2.3.3.1 Single Slave Acknowledge."). The device should be notified of detected errors.

5.3.9.3.1 Cycle Type Sequence Errors.

Every Bus Interface shall differentiate each bus cycle as belonging to one of the bus states listed in Table 5-8. Each slave in a sequence shall determine if the Cycle Type, as indicated by the CT lines, follows a legal sequence as defined under "5.2 GENERAL REQUIREMENTS.." The differentiation of legal and illegal cycle type sequences shall be by comparing the set of bus states that are defined as scheduled states for the current sequence to the actual sequence of symbols received on the CT lines. Table 5-36 shows the required response of a slave to received CT symbols (top row) verses scheduled bus states (left column). Definitions of the required responses are given in Table 5-37.

Table 5-36. Slave Response to Cycle Type Sequence Deviations

	CYCLE TYPE (CT) LINE SYMBOLS							
Scheduled Bus State	I	V	H0	H	D	A	S	AB
I	-	1	2	2	2	2	2	2
V0	2	-	error	error	error	error	error	error
V1, V2, V3	error	-	error	error	error	error	error	error
VZ0 .. VZ3	error	-	error	error	error	error	error	error
H0	4	error	-	error	error	error	error	5
H1 .. H9, HZ	error	error	error	-	error	error	error	5
HA0 .. HA4	error	error	error	error	error	-	error	5
HAZ	error	error	error	error	error	-	error	error
D	error	error	error	error	-	error	3	5
DZ	error	error	error	error	-	error	error	5
DA0 .. DA4	error	error	error	error	error	-	error	5
S0 .. S2	error	error	error	error	error	error	-	5
AB0 .. AB3	error	error	error	error	error	error	error	-

Table 5-37. Cycle Type Deviation Response Definitions

-	Proceed with normal operation.
error	Respond to Cycle Type sequence error as defined in "5.3.9.3.1 Cycle Type Sequence Errors"
1	Monitor vie process (too late to contend).
2	Ignore CT symbol, expect I cycle
3	Suspend message; refer to "5.3.5.1 Suspend.."
4	Expect I cycle (End-Of-Tenure). Set master priority = unknown.
5	Abort message; refer to "5.3.5.2 Abort.."

During Vie, a Bus Interface that detects an illegal CT sequence shall post NAK on the AS lines two cycles after each occurrence of the illegal Cycle Type. Modules which do not win the Vie sequence shall set master priority unknown as described in "5.3.3.1 Vie Sequence.."

If a Cycle Type other than H0 is received as the next Cycle Type following completion of a message sequence, Vie or Abort; each module shall assume that no HWA has been transmitted and, therefore, there is no match for that module's slave ID.

A Bus Interface that is a Slave in a sequence and has detected that the Cycle Type symbols have not followed a legal sequence shall post NAK for every such occurrence. The NAK shall be posted on the second cycle after the cycle that deviated from the legal sequence. If the illegal sequence leads to a condition in which the module cannot determine, with certainty, that the module should be a Slave, the Bus Interface shall not drive any line, other than Bus Request, when legal and required, until a valid H0 or Idle cycle is detected.

5.3.9.3.2 Acknowledge Set, Wait and Bus Request Sequence Errors.

Bus Interfaces shall detect Acknowledge Set, Wait and Bus Request sequence errors. Responses to these errors shall be as defined in Table 5-38. Modules shall not post NAK on the AS lines in response to AS, Wait nor Bus Request sequence errors.

Table 5-38. Sequence Error Response

SIGNAL LINE GROUP	SEQUENCE ERROR RESPONSE
Acknowledge Set	During vie, if not the winner set master priority unknown.
Wait	Assume Wait is not asserted.
Bus Request	Assume Bus Request is not asserted.

5.3.9.4 Semantic Errors.

5.3.9.4.1 Header Semantic Errors.

Each Bus Interface shall detect any error in information transfer that has protocol significance. Table 5-39 lists and defines the errors in this category. In response to these errors, the Bus Interface shall post NAK on the AS lines within two cycles after the error is detected and not later than two cycles after the end of the message or partial message. The Bus Interface shall also log the error in the Acknowledge Word by making the bit named in Table 5-39 a logic 1. The device should be notified that an error was detected.

Table 5-39. Semantic Errors

SEMANTIC ERROR	ERROR DEFINITION
Protect ('Protect Error' bit)	A Write operation has been attempted on a write protected Bus Interface register.
Command ('Command Error' bit)	Header Word A has been received with a reserved code in the AT or MSG Type fields. OR Header Word A has been received with a broadcast slave ID and a single slave Message Type. OR Header Word A has been received and the master's priority is unknown. OR A Tenure Pass Message Header Word A has been received with bits <7..5> asserted or AT not equal to 000 or F asserted or Header Word B bits <4..0> are not equal to the MID in HWA. OR A Type 16 module has been selected and the F bit in HWA is equal to 1. OR A Bus Interface Message Header Word A has been received with F asserted.
Resource not Present ('Resource Not Present Error' bit)	A resource or capability has been addressed that is not implemented. OR A Tenure Pass Message has selected a Feature SO module as the slave. OR A non-existent or reserved Bus Interface register has been addressed.
Device ('Device Error' bit)	Module's device has detected an error attempting to perform bus related operation during the current message.

5.3.9.4.2 Header And Data Acknowledge Semantic Errors.

A bus master Bus Interface shall report semantic errors in the Header and Data Acknowledge Word to the device. The semantic errors which shall be reported are

1. the Acknowledge Word slave MID does not match the physical slave ID transmitted in Header Word A,

2. the Acknowledge Word Type (AWT) field of the Acknowledge Word is not valid for the current Acknowledge cycle,

3. a Class ED master module receives an Acknowledge word with the "Correctable Line Error" bit set to logic 1 and

4. an Acknowledge word which has the "Protect Error" bit set to logic 1 is received when the MSG Type specified in HWA was not Bus Interface Message.

5.3.9.5 Diagnostics.

5.3.9.5.1 On-Line Testing.

On-Line testing of the PI-bus shall be accomplished through the Error Detection and/or Error Correction capability provided by the standard message sequences and protocol defined in previous sections of this specification.

5.3.9.5.2 Off-Line Testing.

Each Bus Interface shall be capable of transmitting and receiving arbitrary bit patterns on all signal lines of the bus (D, DC, CT, CTC, AS, W and BR) in parallel. Multiple clock cycles may be used to establish and read each pattern. Control and coordination of this test shall be through an alternate path independent from the PI-bus under test. The mechanism that coordinates the test should:

1. provide the line patterns to the Bus Interface transmitting the test pattern,

2. determine when the transmit pattern is stable,

3. read the received patterns from the receiving Bus Interface(s) and

4. analyze the patterns for correctness.

The mechanism that controls the test should apply patterns that guarantee detection of 1) a failed line stuck at zero or one and 2) a short between any two lines for any path from the signal line latch of the transmitting Bus Interface to the signal line latch of the receiving Bus Interface(s).

The transmitting Bus Interface shall be capable of being placed in the

off-line test mode, accepting patterns for transmission and transmitting the patterns. Once established, test patterns shall not be changed until a change is commanded by mechanism that coordinates the test.

The receiving Bus Interface, when in the off-line test mode, shall be capable of being commanded to clock in the test pattern from the bus and transfer that received pattern to the controlling device.

VHSIC Phase 2 INTEROPERABILITY STANDARDS

TM-Bus SPECIFICATION

August 31, 1986

Version 1.2

IBM

Honeywell

TRW

Copyright 1985

IBM Honeywell TRW

Copying in printed form is permitted without payment of royalty provided that (1) each reproduction is done without alteration, (2) each reproduction is complete and (3) the copyright notice is included.

CONTENTS

Section	Title	Page
1	SCOPE	1
1.1	Scope	1
1.2	Purpose	1
1.3	Intended Application	1
2	APPLICABLE DOCUMENTS	1
2.1	Government Documents	1
2.2	Non-Government Documents	1
3	DEFINITIONS	1
3.1	Item Definition	1
3.2	Term Definitions	2
4	PHYSICAL LAYER	4
4.1	Introduction	4
4.2	Line Definition	4
4.2.1	Nomenclature	4
4.2.2	TM-Bus Signal Definition	5
4.2.2.1	TM-Bus CLOCK Signal Definition	5
4.2.2.2	TM-Bus MASTER DATA Signal Definition	5
4.2.2.3	TM-Bus SLAVE DATA Signal Definition	5
4.2.2.4	TM-Bus CONTROL Signal Definition	5
4.2.2.5	TM-Bus Addressing	6
4.3	ELECTRICAL REQUIREMENTS	7
5	DATA LINK LAYER	8
5.1	Introduction	8
5.2	Operation	8
5.2.1	TM-Bus States	9
5.2.2	Message/Response Packet Descriptions	11
5.2.2.1	DATA PACKETS	11
5.2.2.2	Packet Parity	11
5.2.2.3	HEADER Packets	11
5.2.2.4	ACKNOWLEDGE Packets	12
5.2.3	Message Protocol	12
5.2.4	Response Protocol	13
5.2.5	Broadcast Capability	18
5.2.6	Multicast Capability	18
5.2.7	TM-Bus SLAVE Status Register	19
5.2.8	TM-Bus Interrupts	20
5.2.9	TM-Bus Contention	21
5.3	Command Definitions	24
5.3.1	Reset SLAVE	24
5.3.2	Initialize Module	24
5.3.3	Read Status Register	24
5.3.4	CONTEND for Bus	24

CONTENTS (continued)

Section	Title	Page
5.3.5	Enable Interrupt	25
5.3.6	Disable Interrupt	25
5.3.7	Multicast Select 0	25
5.3.8	Multicast Select 1	25
5.3.9	Multicast Select 2	25
5.3.10	Multicast Select 3	25
5.4	TM-Bus Synchronization/Initialization	26
5.5	TM-Bus Mastership	26
5.6	TM-Bus Error Handling	27
5.7	TM-Bus Testing	27
6	NOTES	28
Appendixes		
10	GLOSSARY	29

CONTENTS (continued)

LIST OF ILLUSTRATIONS

Figure	Title	Page
Figure 1	Conceptual Model Of Bus And Modules	2
Figure 2	TM-BUS Signals	6
Figure 3	TM-Bus Design Parameters and Characteristics	8
Figure 4	TM-BUS STATES	9
Figure 5	TM-Bus State Diagram	10
Figure 6	HEADER PACKET	11
Figure 7	ACKNOWLEDGE PACKET	12
Figure 8	MESSAGE PROTOCOL	13
Figure 9	DATA TRANSFER TIMING	14
Figure 10	SLAVE RESPONSE PROTOCOL (A)	15
Figure 11	SLAVE RESPONSE PROTOCOL (B)	16
Figure 12	SLAVE RESPONSE FLOW	17
Figure 13	SLAVE Status Register	20
Figure 14	INTERRUPT TIMING	21
Figure 15	CONTEND SEQUENCE	23
Figure 16	Standard Commands	26

1 SCOPE

1.1 Scope. This specification establishes the electrical, functional and performance requirements for the set of signal lines that constitute the Test and Maintenance Bus (TM-Bus).

1.2 Purpose. The purpose of this standard is to establish requirements for the TM-Bus and facilitate interoperability of modules which use the TM-Bus.

1.3 Intended Application. The TM-Bus is intended as a serial path for test and maintenance control and data information.

2 APPLICABLE DOCUMENTS

2.1 Government Documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superceding requirement.

- VHSIC Phase 2 INTEROPERABILITY STANDARDS PI-Bus SPECIFICATION

2.2 Non-Government Documents. The following documents of the exact issue shown form a part of this specification to the extent specified herein. In the event of conflict between the documents referenced herein and the contents of this specification, the contents of this specification shall be considered a superceding requirement.

- None.

3 DEFINITIONS

The definitions listed here shall apply to the TM-bus and TM-bus modules.

3.1 Item Definition. The TM-bus is a linear, multi-drop communications medium which transfers bit serial data between a 'MASTER' module and up to 32 'SLAVE' modules residing on a single backplane.

TM-bus modules implement the TM-bus protocol and meet all requirements of

this specification.

Figure 1, illustrates the TM-bus and TM-bus modules. Conceptually, each module consists of a device which performs the application specific function of the module and a bus interface which implements the TM-bus MASTER-SLAVE communications protocol.

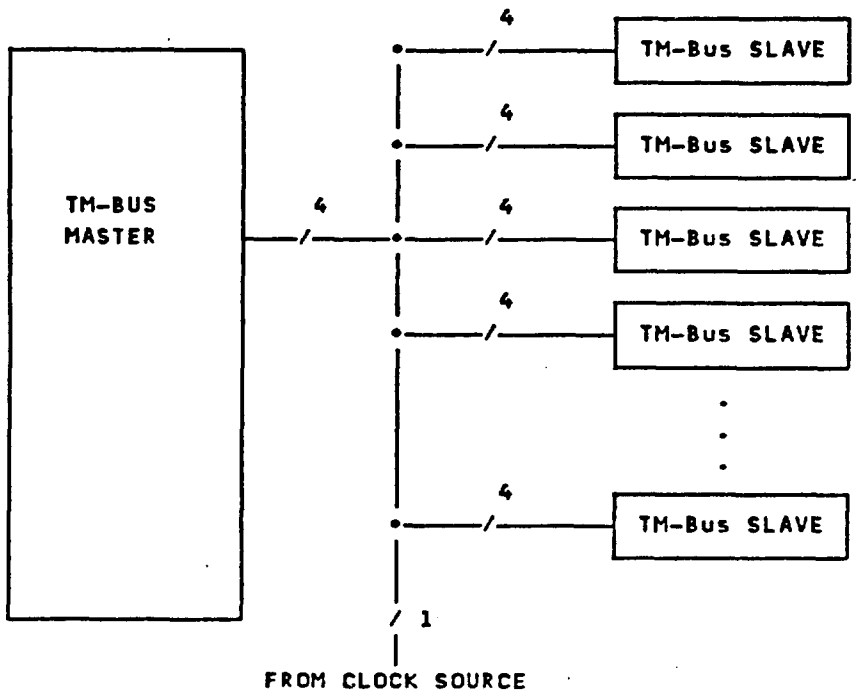


Figure 1. Conceptual Model Of Bus And Modules

3.2 Term Definitions.

active bus interface	A bus interface that is connected to the bus, and is currently capable of (and not inhibited from) participating in bus transactions.
assert	The action of changing the state of a bus signal line from released, logic 0, to asserted, logic 1, or of ensuring that the line remains in the asserted state.

asserted	The logic 1 state of a bus signal line. The least positive of the two states of a bus signal line.
backplane	A motherboard comprising wiring for the bus and connectors to the modules attached to the bus.
broadcast	A mode of operation where the bus MASTER transmits data to all SLAVE modules during a single sequence.
bus MASTER	The module in control of the bus.
contend	When a bus SLAVE module(s) is/are actively vying for the attention of the bus MASTER.
device	The portion of a module, excluding the bus interface, which does the application dependent function of the module.
header	A sequence identifying a bus command, the SLAVEs participating in any commanded sequence and additional information delimiting the scope of the command.
linear bus	A bus with a single shared medium segment.
message	A set of sequences starting with a header and terminating when all bus actions indicated by that header have been performed.
module	An entity which is addressable via the bus and has a single bus connection.
module address	A pointer which uniquely identifies a module.
multicast	A mode of operation where the MASTER may transmit data to more than one SLAVE during a single sequence.
packet	A unit of data which is 17 bits, a 16 bit word plus 1 parity bit.
release	The action of ceasing to assert a logic 1 on a bus signal line. The action of releasing a signal line produces a change in the state of the signal line only if no module is asserting that signal.

released	The logic 0 state of a bus signal line produced when no module asserts the signal associated with that line. The more positive of the two states of a bus signal line relative to the 0 Volt logic reference.
response	A set of sequences sent by the SLAVE as a result of a message being sent by the MASTER.
sequence	An indivisible transaction comprising a number of transfers performing one intended function.
SLAVE	A module which does not have control of the bus but which is selected by the MASTER to participate in a sequence.
sub-address	A pointer to elements within an addressable module.
transfer	A set of elemental operations on the bus which result in the communication of bit serial datum units between the current bus MASTER and the selected SLAVE(s). A serial datum unit is 1 bit. See sequence.
word	An ordered set of 16 bits operated on as a unit. The most significant bit is labeled bit 16 and the least significant bit is labeled bit 1.

4 PHYSICAL LAYER

4.1 Introduction. The physical layer of the TM-Bus is specified herein. The lines required to implement the TM-Bus are defined, the electrical characteristics of the modules and backplane are specified and timing definitions are presented. The bus interface facilities which are accessible to a bus MASTER over the bus are also defined.

4.2 Line Definition. The TM-Bus signal, clock and module identification lines are defined in this section.

4.2.1 Nomenclature. Lines shall be designated by name. When a set of related bits are represented by the same name, the bits within the set shall be differentiated by number with the least significant bit numbered 0. All fields shall be referred to by their bit position within a data word trans-

ferred over the TM-bus. The nomenclature for single bits shall be the bit number enclosed in < >. The nomenclature <m..n> shall be the abbreviation for the set of bits m to n inclusive, where m and n are the most and least significant bits respectively.

4.2.2 TM-Bus Signal Definition. There shall be four (4) signal types that make up the TM-Bus as shown in Figure 2 on page 6. All bus signals shall use negative logic, i.e. the logic '1' state (or asserted state) is the lowest voltage level on the bus and the logic '0' (or released state) is the higher voltage level on the bus.

4.2.2.1 TM-Bus CLOCK Signal Definition. The TM-Bus CLOCK signal shall be a single phase clock. The TM-Bus interface shall support the full range of clock frequencies from zero (0) Hz to 6.25 MHz. All control and data transfer operations shall be synchronized with the TM-Bus CLOCK signal. All data and commands shall be placed on the TM-Bus on the high to low transition of the clock and latched-in on the next high to low transition.

4.2.2.2 TM-Bus MASTER DATA Signal Definition. The TM-Bus MASTER DATA signal shall be a single uni-directional line used to transmit device addresses, instruction data, and/or scan data from the MASTER to the SLAVE(s). The MASTER DATA line is also used in conjunction with the CONTROL line to indicate bus states (see Section "5.2.1 TM-Bus States" on page 9).

4.2.2.3 TM-Bus SLAVE DATA Signal Definition. The TM-Bus SLAVE DATA signal shall be used to transmit acknowledgements, data, and/or interrupts from the SLAVE(s) to the MASTER. The TM-Bus SLAVE DATA line shall support a wired-OR configuration.

4.2.2.4 TM-Bus CONTROL Signal Definition. The TM-Bus CONTROL signal shall be a single uni-directional line from the MASTER to the SLAVES(s). When the CONTROL line is asserted the bus is placed in the DATA TRANSFER state. When CONTROL is released, the bus is in the PAUSE or IDLE state.

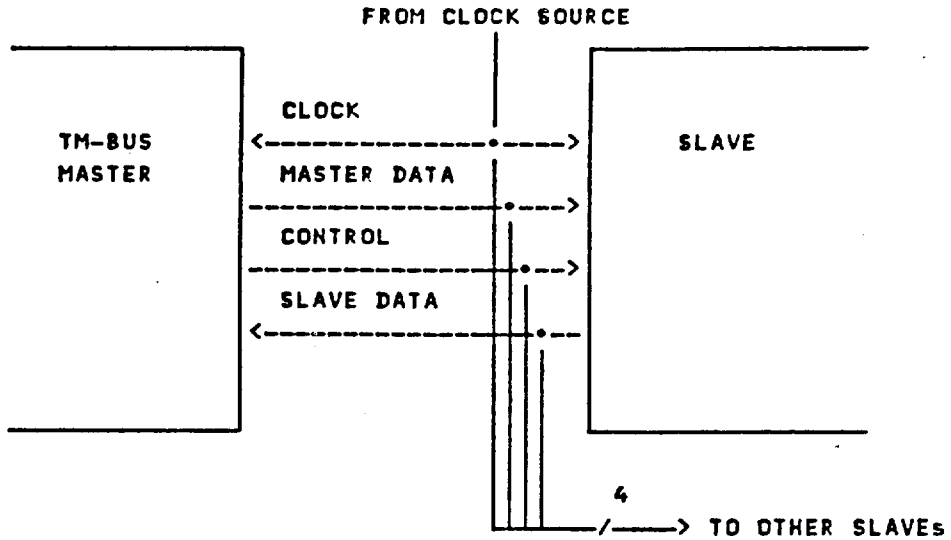


Figure 2. TM-BUS Signals

4.2.2.5 TM-Bus Addressing. Each TM-Bus SLAVE is addressed by an eight bit address field. This address shall be sent in the HEADER packet containing the five (5) bit module address (bits <16..12>) and the three (3) bit sub-address (bits <11..9>), (see Figure 6 on page 11).

The five-bit module address field in the HEADER shall be compared to five Module IDentification (MID) inputs to determine if the SLAVE is being addressed. As a minimum, each SLAVE shall also have a Module Identification Parity (MIP) input that shall be set such that the modulo two sum of the five MID inputs and the MIP input equals one (1). (Note: the asserted state of each input is a logical one). When used in conjunction with the VHSIC Phase 2 PI-Bus, it is recommended that each TM-Bus SLAVE module have its MID and MIP inputs hardwired to the backplane of the TM-Bus (see section "4.2.4 Module Identification" of the PI-Bus Specification). If an unrecoverable error occurs on the MID inputs, the TM-Bus SLAVE shall not execute any commands and shall release the SLAVE DATA line.

Comparison of the three (3) sub-address bits from the HEADER packet against Sub-address IDentification (SID) inputs is optional.

Module addresses '0' through '30' have a maximum of eight (8) subaddresses. Address '31' is limited to three (3) subaddresses ('F8', 'F9', and 'FA' HEX) due to restrictions of broadcast and multicast commands. See Section "5.2.5

Broadcast Capability" on page 18 and Section "5.2.6 Multicast Capability" on page 18 for details.

4.3 ELECTRICAL REQUIREMENTS. Electrical characteristics for the TM-Bus backplane and modules shall be as specified in the most recent version of the VHSIC Phase 2 PI-Bus Specification, Section "4.3 ELECTRICAL REQUIREMENTS".

5 DATA LINK LAYER

5.1 Introduction. The Test and Maintenance Bus (TM-Bus) shall be the channel for control and data information flow between a maintenance controller and modules within a system. The module in control of the TM-Bus shall be referred to as the MASTER and all other modules on the TM-Bus shall be referred to as SLAVEs. The information transferred and the scheduling of data and commands is system dependent and is not addressed in this specification. Figure 3 on page 8 summarizes the TM-Bus design parameters and characteristics.

o Performance Characteristics	o Protocol Characteristics
- 6.25 MHz clock (Typical)	- 8 reserved address bits
- 4 pin bus signals	- 32 module addresses (maximum)
- Synchronous Operation	- 8 sub-addresses per module address
- Two Data Lines	- Multi-drop Configuration
- SLAVE status register	- Interrupt Capability

Figure 3. TM-Bus Design Parameters and Characteristics

5.2 Operation. Messages transmitted by the MASTER shall consist of a command HEADER packet, and optional DATA packets. If required, the SLAVE shall respond by acknowledging the command and/or transmitting any DATA packets requested. The SLAVE shall only transmit packets when requested to do so by the MASTER. The SLAVE shall indicate interrupts as specified in Section "5.2.8 TM-Bus Interrupts" on page 20. All data shall be transmitted MSB first.

The following figures and paragraphs describe the operation of the four line serial TM-Bus in detail.

5.2.1 TM-Bus States. There are three possible bus states as shown in Figure 4. The IDLE state indicates that data shall not be

CONTROL	MASTER DATA	STATE
0	0	IDLE/INTERRUPT (End of Message (EOM))
0	1	PAUSE/INTERRUPT
1	0	DATA TRANSFER
1	1	DATA TRANSFER

Figure 4. TM-BUS STATES

transferred over the bus but interrupts from the SLAVEs are allowed over the SLAVE data line. The PAUSE state shall be used between packets during a message transfer to allow SLAVEs to interrupt the MASTER. The DATA TRANSFER state indicates that data shall be transferred over the MASTER data line or the SLAVE data line or both. Figure 5 on page 10 shows the state diagram for the TM-Bus.

Packets in a transmission may be separated by a variable number of PAUSE states (typically from 0 to 5, system requirements may dictate a higher number). The end of a message shall be signified by a return to the IDLE state.

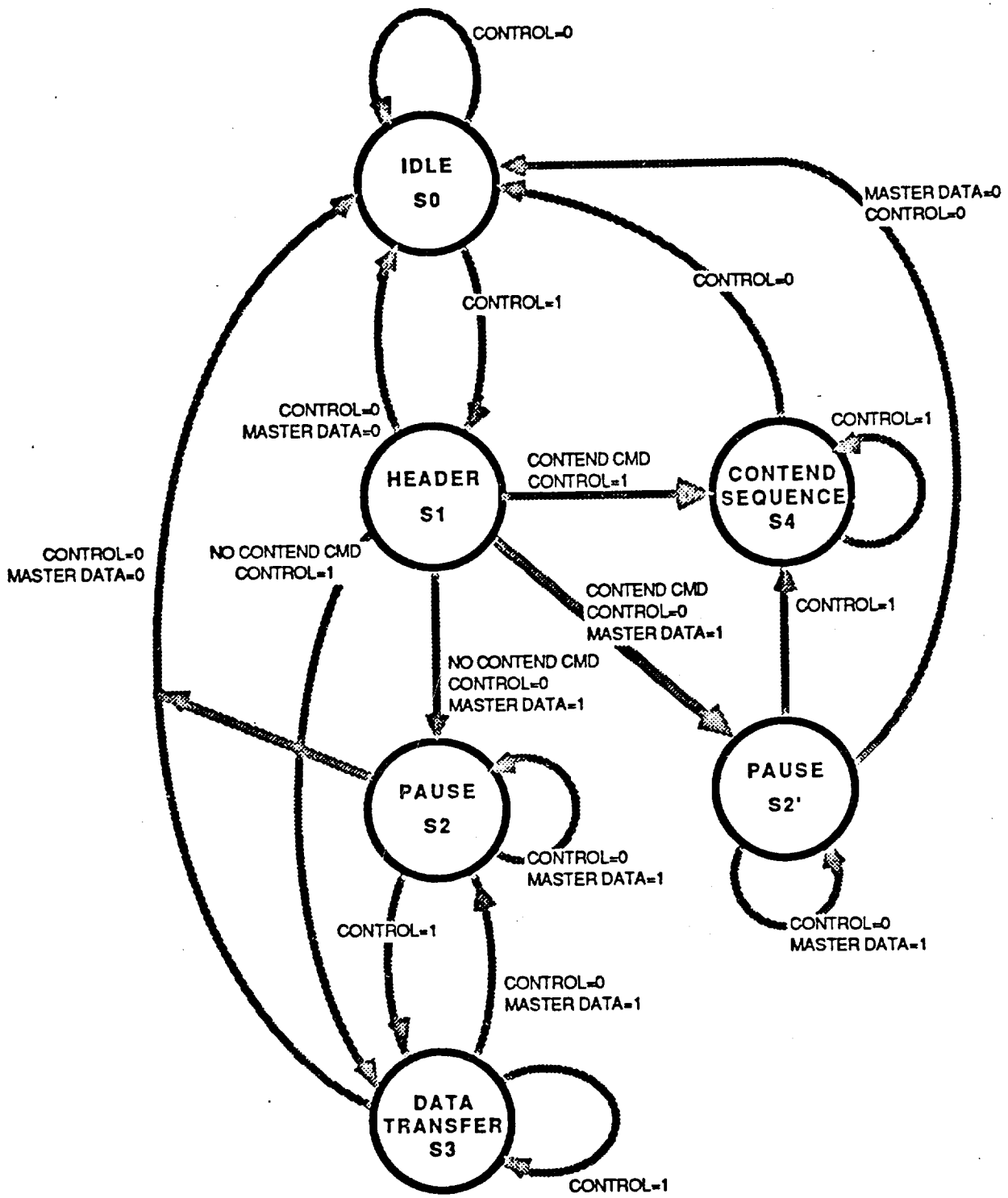


Figure 5. TM-Bus State Diagram

5.2.2 Message/Response Packet Descriptions. All messages sent by the MASTER shall consist of a HEADER and up to 256 DATA packets. Responses (from the SLAVE) shall consist of an optional ACKNOWLEDGE packet and/or up to 256 DATA packets. To allow flexibility, the number of DATA packets contained in a response is determined by user definable commands (with a maximum of 256).

5.2.2.1 DATA PACKETS. The DATA packet contains sixteen (16) data bits (bits <16..1>) and one packet parity bit (bit <0>). The contents and format of the data bits are not specified. Data shall be sent MSB (bit 16) first.

5.2.2.2 Packet Parity. Bit 0 of each packet shall contain one packet parity bit. The parity shall be odd parity such that the modulo 2 sum of a data packet (bits <16..0>) = 1. The parity bit shall be transmitted last as the LSB.

5.2.2.3 HEADER Packets. Figure 6 shows the format for the HEADER packet which includes the SLAVE address and command fields. The SLAVE address field is eight (8) bits in length (bits <16..9>). The SLAVE command field is seven (7) bits in length (bits <8..2>). The ACKNOWLEDGE REQUEST field is one bit in length (Bit <1>).

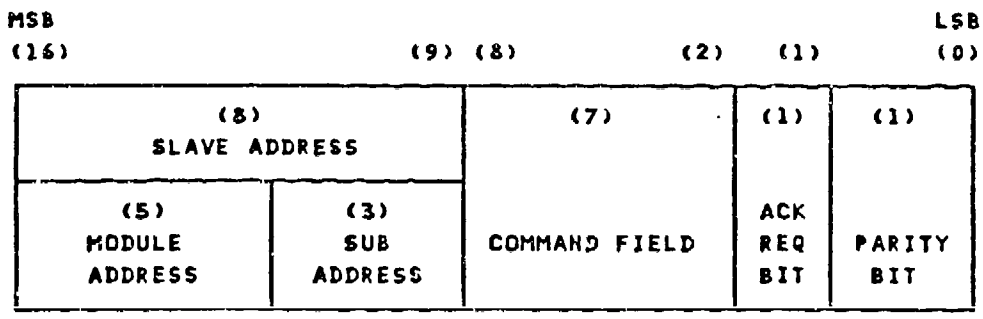


Figure 6. HEADER PACKET

The standard commands are defined in Section "5.3 Command Definitions" on page 24. If the ACKNOWLEDGE REQUEST bit (bit <1> of the HEADER) is asserted, the SLAVE shall respond with an ACKNOWLEDGE packet. Bit 0 is the packet parity bit.

5.2.2.4 ACKNOWLEDGE Packets. Figure 7 on page 12 shows the format for the ACKNOWLEDGE packet which includes the SLAVE address and status fields. The SLAVE address field is eight (8) bits in length (bits <16..9>) and contains the address of the responding SLAVE. The status field is also eight (8) bits in length (bits <8..1>) and contains the data residing in the SLAVE Status Register. Bit 0 is the parity bit.

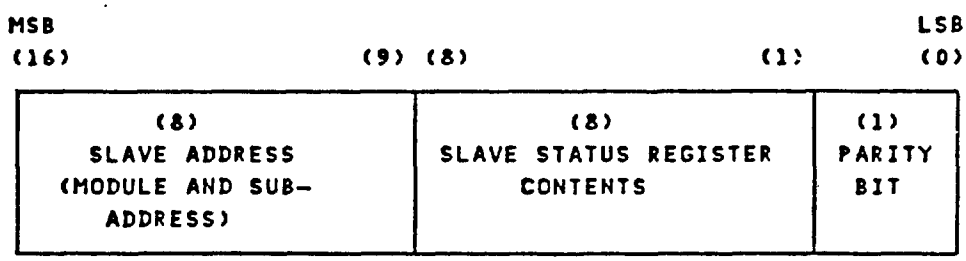


Figure 7. ACKNOWLEDGE PACKET

5.2.3 Message Protocol. Message transmissions from the MASTER to the SLAVE shall be as shown in Figure 8. A message transmission shall begin by moving from the IDLE state to the DATA TRANSFER state (the CONTROL line being asserted) at which time the HEADER packet is transmitted. The CONTROL line shall be asserted for the duration of each packet transmission. The IDLE/EOM state shall be indicated at the end of a a transmission by releasing the CONTROL line (MASTER DATA line shall be released in the IDLE/EOM state). The PAUSE state shall be indicated between packets by releasing the CONTROL line (with MASTER DATA line asserted). If the message is more than one packet in length, then the CONTROL line shall be asserted during each additional packet transmission. Optional PAUSE states are permitted between packets within a single message. The maximum number of PAUSE states shall be determined by system requirements, and the minimum number of PAUSE states is zero.

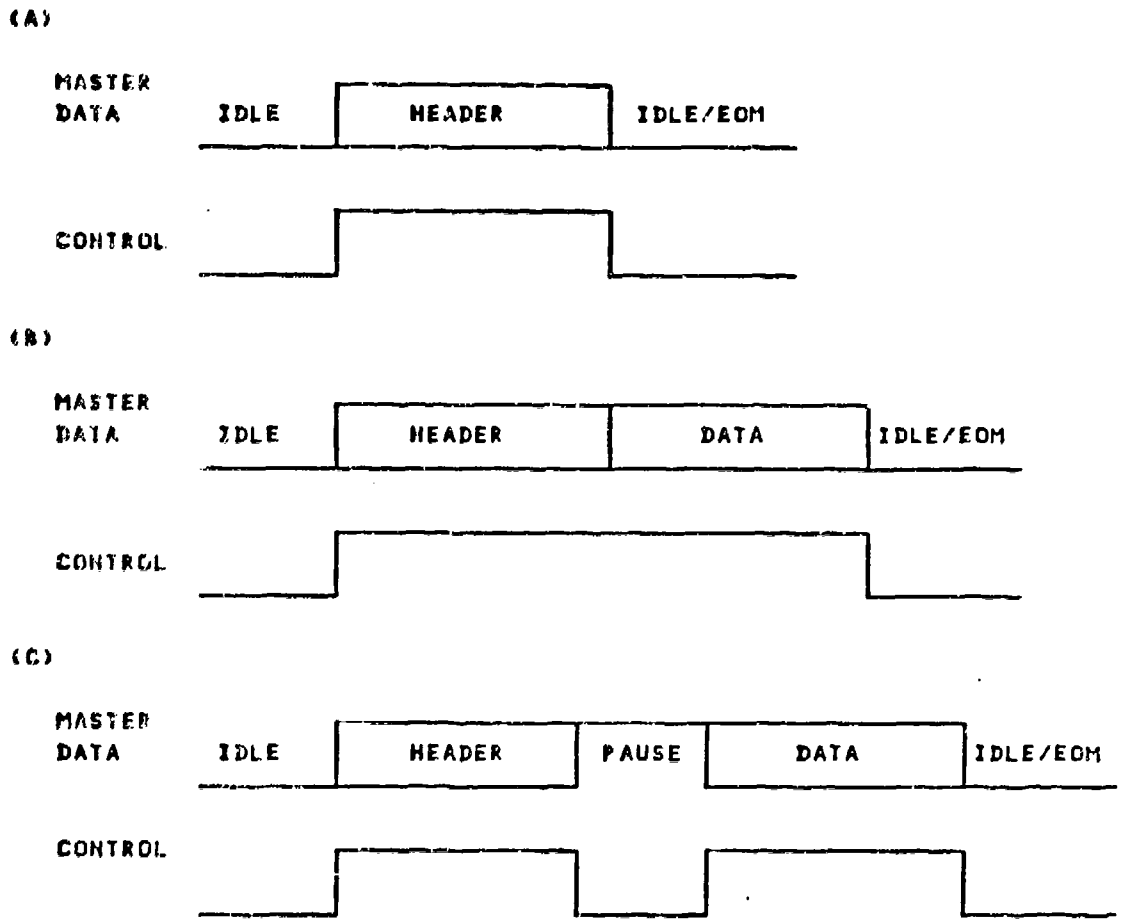


Figure 8. MESSAGE PROTOCOL

5.2.4 Response Protocol. All states on the SLAVE DATA line including PAUSE, IDLE, and packet transmissions, shall be synchronous to the MASTER DATA and CONTROL lines with a two clock cycle delay as shown in Figure 9 on page 14. This delay is required so that the SLAVE can receive and react to state transitions on the MASTER DATA and CONTROL lines. The SLAVE DATA line shall begin a packet transmission with the CONTROL line asserted, following receipt of a HEADER packet as shown in Figure 10 and Figure 11 with the two cycle delay described above. The Flowchart of Figure 12 on page 17 shows the flow diagram of the SLAVE response.

If the ACKNOWLEDGE bit is asserted in the HEADER, the addressed SLAVE shall respond with an ACKNOWLEDGE packet during the next packet transmission period. After the optional ACKNOWLEDGE packet, any DATA packets required by the decoded command shall be transmitted as shown in Figure 11 on page 16. The SLAVEs shall not respond with an acknowledge during broadcast or multicast operations.

If the message to the SLAVE contains DATA packets as shown in Figure 11 and the acknowledge bit is asserted, then the SLAVE shall respond by sending an ACKNOWLEDGE packet during the next period that data may be sent over the SLAVE DATA line. After the optional ACKNOWLEDGE packet, any DATA packets required by the command shall be transmitted as shown.

As shown in Figure 11 on page 16 data transfer may occur simultaneously on the MASTER DATA line and the SLAVE DATA line. This simultaneous transfer is dependent on the command received by the SLAVE. None of the standard com- ··all require simultaneous transmissions.

Interrupt Handling protocol is described in Section "5.2.9 TM-Bus Contention" on page 21.

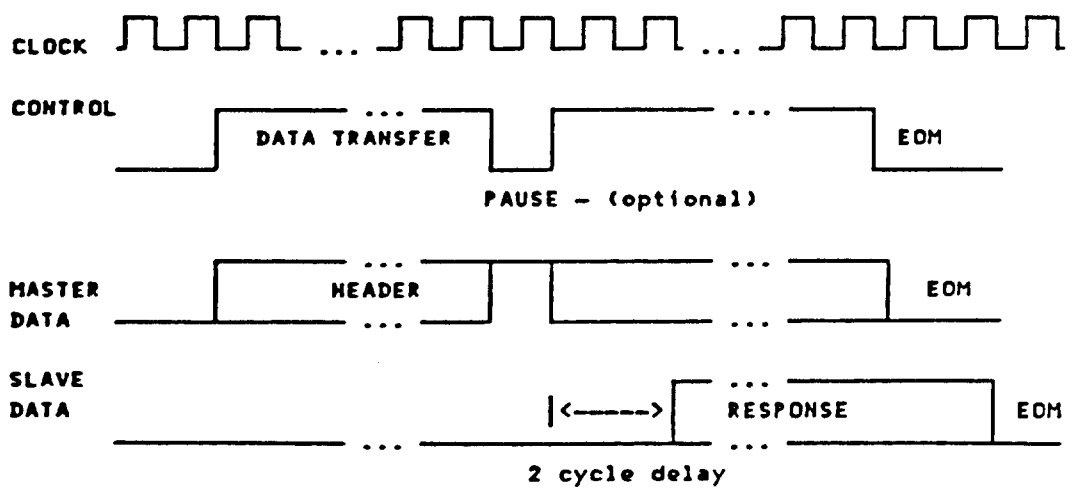


Figure 9. DATA TRANSFER TIMING

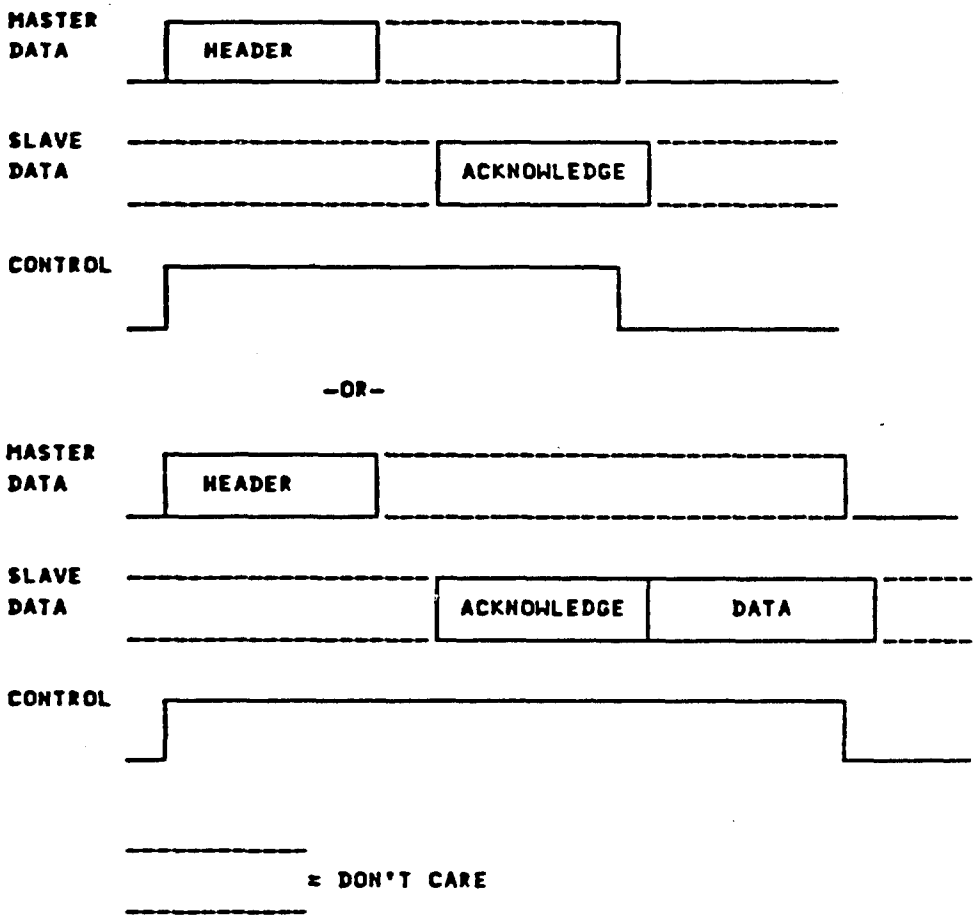


Figure 10. SLAVE RESPONSE PROTOCOL (A)

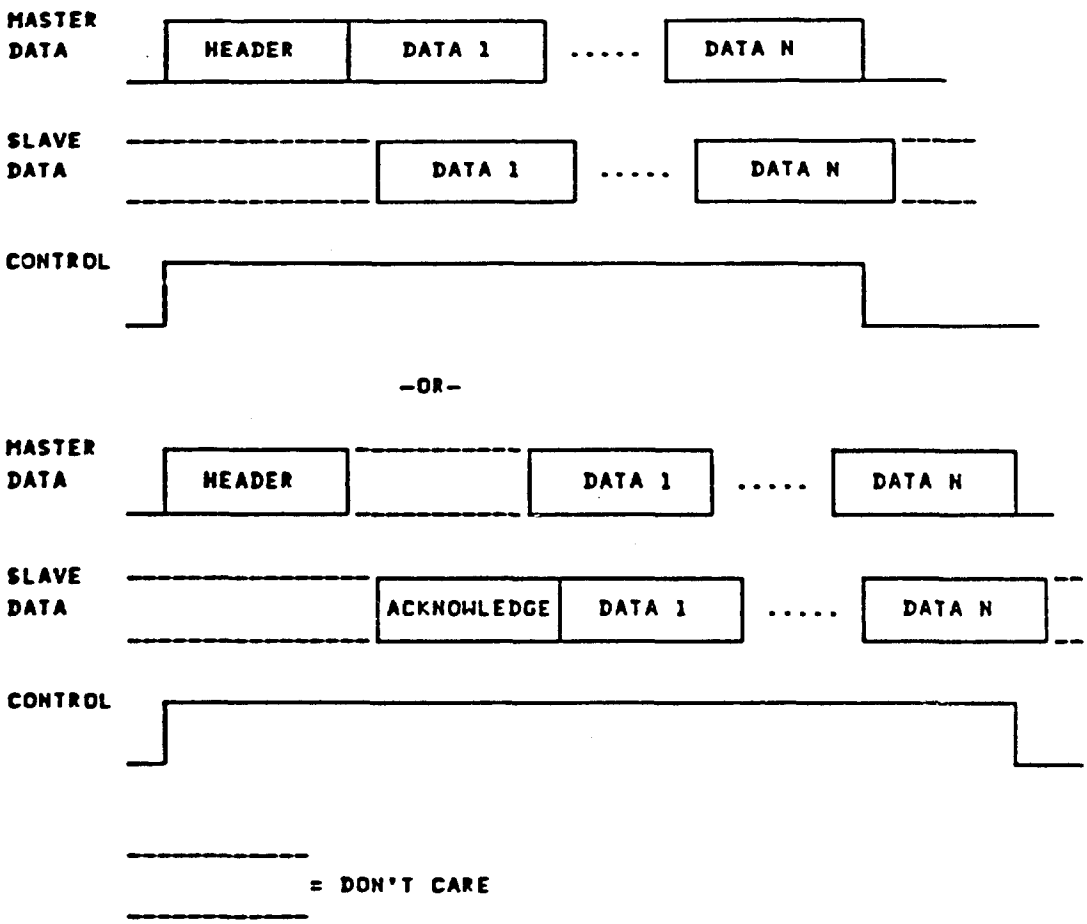


Figure 11. SLAVE RESPONSE PROTOCOL (B)

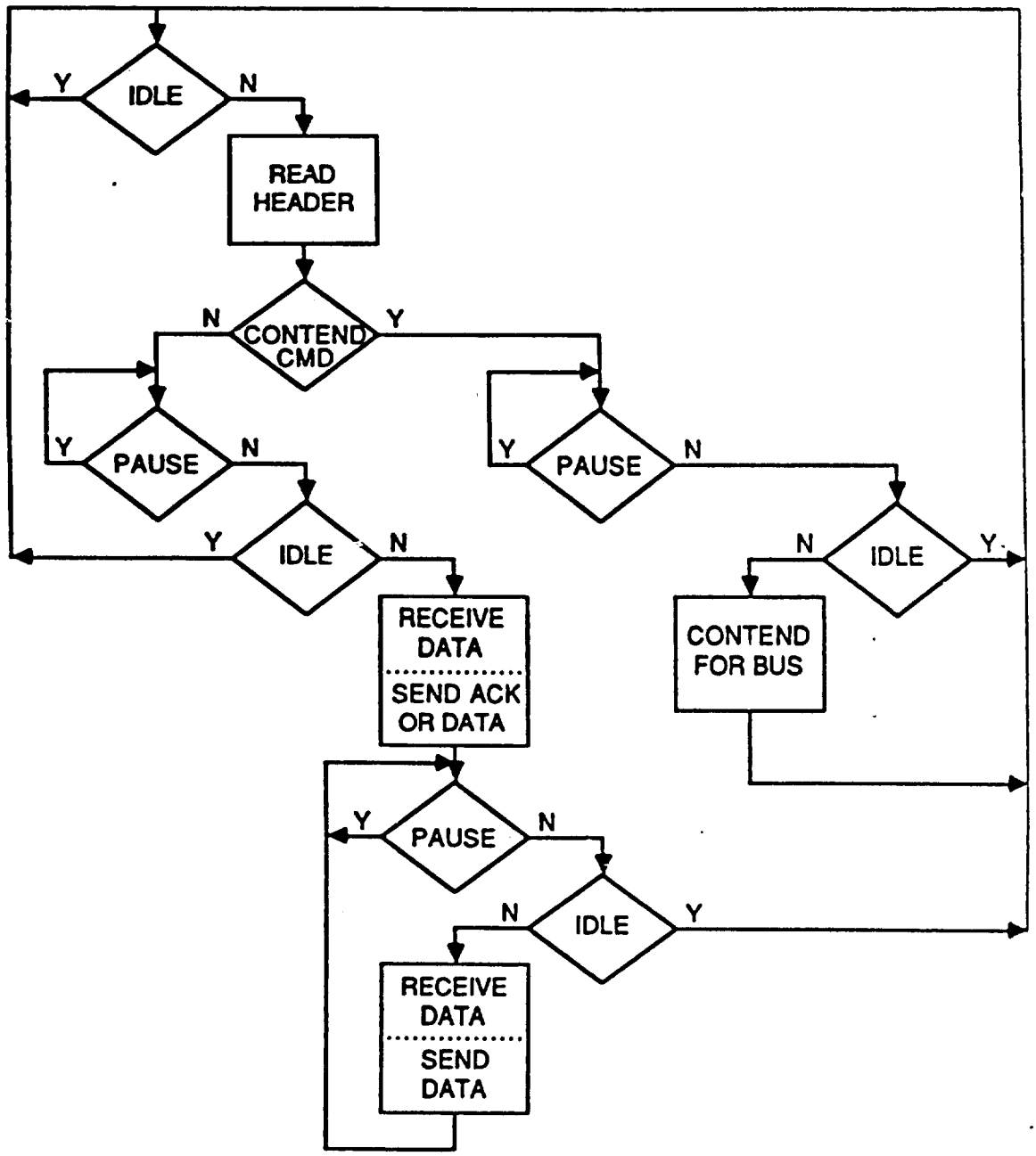


Figure 12. SLAVE RESPONSE FLOW

5.2.5 Broadcast Capability. The MASTER shall have the capability to broadcast to all SLAVEs by setting the SLAVE address field equal to ('FB' HEX). All SLAVEs shall recognize this address in addition to their normal module address and/or sub-address. SLAVEs shall indicate correct receipt of a broadcast command (HEADER packet) by asserting the Broadcast/Multicast Received bit in the SLAVE status register. The Broadcast/Multicast Received bit shall be released if the broadcast command was not received correctly or the SLAVE was busy during a broadcast operation, such that it could not execute the TM-Bus command. See sections "5.3.1 Reset SLAVE" on page 24, "5.3.3 Read Status Register" on page 24, and "5.6 TM-Bus Error Handling" on page 27 for further discussions of the broadcast/multicast received bit. Note that a SLAVE shall have the capability to execute the standard commands regardless of their busy state. SLAVEs shall assert the SLAVE Busy or the Bus Error bit in the SLAVE status register if a broadcast command is not received properly. SLAVEs shall not transmit any response packets over the SLAVE DATA line in response to broadcast operations except during CONTEND commands. SLAVEs may issue interrupts during a broadcast operation (see section "5.2.8 TM-Bus Interrupts" on page 20).

5.2.6 Multicast Capability. The MASTER shall have the capability to multicast to any number of SLAVEs. SLAVEs shall belong to one of four multicast groups (00, 01, 10, 11). SLAVEs shall be selected for these groups by sending one of the four Multicast Select commands. The SLAVE status register shall maintain two bits indicating which of the four possible multicast groups the SLAVE belongs to. When the TM-Bus is reset, via a Reset SLAVE command, SLAVEs are set to membership in group '00'. A SLAVE shall remain a member of a multicast group until a Multicast Select Command (or Reset SLAVE command) is sent to change the SLAVE to another group.

Four addresses shall be recognized by SLAVEs as valid multicast addresses. Address ('FC' Hex), ('FD' HEX), ('FE' HEX) and ('FF' HEX) shall be used for multicast groups '00', '01', '10', and '11' respectively. SLAVEs shall indicate correct receipt of a multicast command (HEADER packet) by asserting the Broadcast/Multicast Received bit in the SLAVE status register. The Broadcast/Multicast Received bit shall be released if the multicast command was not received correctly or the SLAVE was busy during a multicast operation, such that it could not execute the TM-Bus command. See sections "5.3.1 Reset SLAVE" on page 24, "5.3.3 Read Status Register" on page 24, and "5.6 TM-Bus Error Handling" on page 27 for further discussions of the broadcast/multicast received bit. Note that a SLAVE shall have the capability to execute the standard commands regardless of their busy state. SLAVEs shall assert the SLAVE Busy or the Bus Error bit in the SLAVE status register if a multicast command is not received properly. SLAVEs shall not transmit any packets over the SLAVE DATA line in response to multicast operations, except during Contend commands. See section "5.3 Command Definitions" on page 24 for details of Multicast Select commands. SLAVEs may issue interrupts dur-

ing a multicast operation (see section "5.2.8 TM-Bus Interrupts" on page 20).

5.2.7 TM-Bus SLAVE Status Register. Each SLAVE shall have a SLAVE Status Register described in Figure 13 on page 20. All bits in the status register shall be considered active when asserted. The Bus Error, Broadcast/Multicast Received, and Event Occurrence bits shall be reset to a logic '0' (released) when the SLAVE's Status Register is read by a Read Status Register command. Resetting the Reserved bit in the Status Register shall be optional when this command is executed. The Bus Error, Broadcast Received, and Event Occurrence bits shall be reset to a released state when a SLAVE wins a contend sequence. Resetting the Reserved bit in the Status Register shall be optional when a SLAVE wins a contend sequence.

Bit	Name	Meaning When Active
8(MSB)	Reserved	Available for user defined status. May be used for address extension.
7	SLAVE Busy	Indicates that the application side of the TM-Bus interface is busy.
6	Event Occurrence	Indicates that an error condition or other predefined condition exists.
5	Broadcast/ Multicast Received	Indicates that the last Broadcast/ Multicast command was properly received.
4	Bus Error	Indicates that a parity error or an illegal command has been detected by the SLAVE.
3	Multicast Select Bit 1	Indicates SLAVE multicast select Mode.
2	Multicast Select Bit 0.	Indicates SLAVE multicast select Mode.
1(LSB)	Interrupt Enabled	Indicates whether the SLAVE may send an interrupt.

Figure 13. SLAVE Status Register

5.2.8 TM-Bus Interrupts. Any SLAVE may signal an interrupt to the MASTER by asserting the SLAVE DATA line for one clock period/cycle during the PAUSE or IDLE states, (when interrupts are enabled) as shown in Figure 14. On receiving an interrupt, the MASTER may service that interrupt by issuing a 'CONTEND for bus' command, checking the error status bits, and taking appropriate action.

The SLAVE shall send an interrupt out over the SLAVE DATA line when the Event Occurrence or Bus Error bits are asserted. The SLAVE shall consider the interrupt condition serviced when the SLAVE wins a contend sequence or the MASTER issues a Read Status Register command to that SLAVE. The SLAVE

shall continue to send the interrupt for one clock period after all subsequent contend sequences, that the SLAVE does not win, until the interrupt is serviced. All interrupts shall be sent only during periods that interrupts are valid on the bus.

If the SLAVE Busy bit is asserted during data transfers (Bus State S3) after the optional ACKNOWLEGE packet is transferred, the SLAVE shall send an interrupt over the bus.

Any SLAVE that is currently addressed shall have the ability to interrupt during PAUSE states within a message. An active SLAVE's interrupt capability shall override the DISABLE INTERRUPT command. The SLAVE shall go back to the state selected by the last DISABLE or ENABLE INTERRUPT COMMAND following completion of a bus transaction.

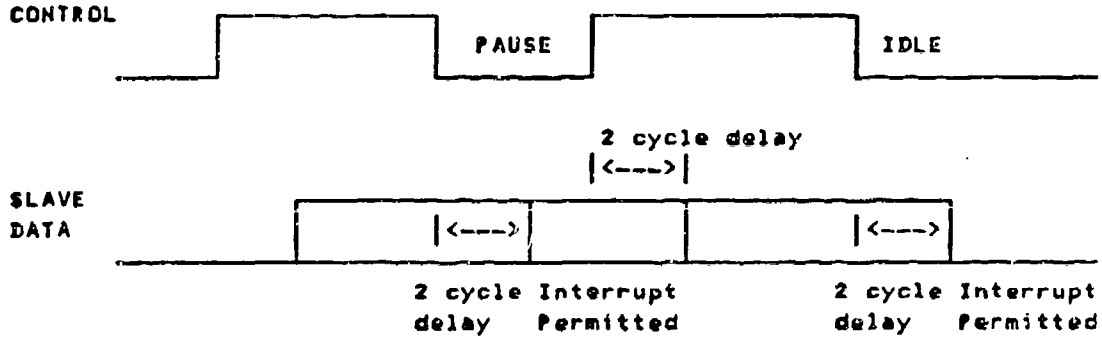


Figure 14. INTERRUPT TIMING

5.2.9 TM-Bus Contention. When the CONTEND for bus command is issued, any number of SLAVEs may then CONTEND for the bus by simultaneously transmitting their ACKNOWLEDGE packet which includes the eight (8) bit SLAVE address. SLAVEs shall participate in the CONTEND command only when they have a predefined event occur (such as an error condition, see Section "5.2.8 TM-Bus Interrupts" on page 20) that caused or would cause the SLAVE to send an interrupt to the MASTER. SLAVEs shall not CONTEND for the bus if their interrupt has been disabled. During the transmission of this packet, the SLAVE shall 'listen' to the SLAVE DATA line and inhibit data transmission if a higher priority address is 'heard'. The highest SLAVE address shall have the highest priority.

The CONTEND sequence requires two clock cycles for the presentation of each SLAVE ACKNOWLEDGE bit, whereby the SLAVEs transmit their most significant address bit first, then 'listen' to the bus for a higher address. If the higher address is not 'heard', then a SLAVE shall continue to alternately transmit and 'listen' until the entire 16 bit SLAVE ACKNOWLEDGE packet has been placed on the bus (in 32 clock cycles) and one bit of parity in the 33rd cycle, or until a higher address is 'heard' on the bus (as shown in Figure 15).

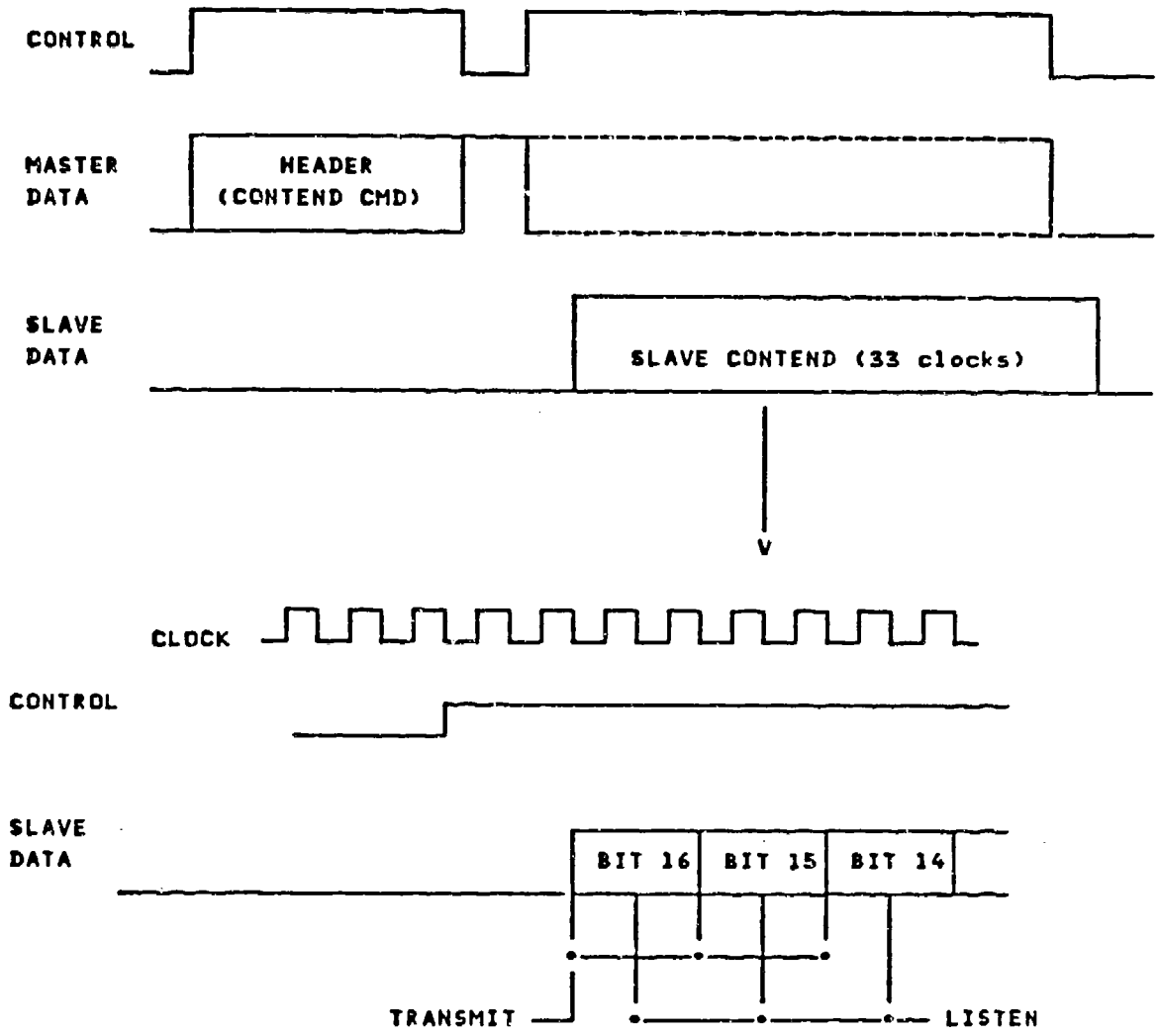


Figure 15. CONTEND SEQUENCE

5.3 Command Definitions. The HEADER commands shown in Figure 16 on page 26 are defined below. There shall be 7 bits allowing 127 SLAVE commands. Commands 0 through 15 shall be standard or reserved commands and the remainder shall be user defined commands. The command ('7F' HEX) with the ACKNOWLEDGE REQUEST bit asserted shall be an illegal command to all SLAVEs. If the '7F' command is detected by a SLAVE, with the ACKNOWLEDGE REQUEST bit asserted, then the SLAVE shall set the Bus Error bit in the SLAVE status register. SLAVEs shall execute all the commands defined below regardless of the state of the BUSY bit in the SLAVE status register. The results of the following commands shall be reflected in the returned ACKNOWLEDGE packet (if ACKNOWLEDGE has been requested) except for the Read Status command, see Section "5.3.3 Read Status Register" on page 24.

5.3.1 Reset SLAVE. This command shall bring the TM-Bus SLAVE(s) to an error-free quiescent state and resets all internal registers, counters, and buffers to a known initial state, such that the SLAVE is capable of receiving and executing commands. When a SLAVE receives the Reset command, all of the SLAVE Status Register bits shall be reset to a released state, the SLAVE's multicast select group mode shall be reset to '00', and SLAVE interrupts shall be disabled.

A broadcast or multicast reset command shall set the Broadcast/Multicast Received bit.

5.3.2 Initialize Module. The application side of the module shall be initialized by resetting or setting required registers to a pre-defined state.

5.3.3 Read Status Register. Upon the receipt of a NON-broadcast/multicast read status command, the SLAVE shall return the ACKNOWLEDGE packet which includes the current eight (8) bit SLAVE Status Register contents. This command shall then reset the Bus Error, Event Occurrence , and Broadcast Received bits to the released state. Resetting the User Defined bit in the SLAVE Status Register shall be optional.

If a broadcast/multicast read status command is received, the SLAVE shall not transmit any response over the SLAVE DATA line. The Bus Error, Event Occurrence, and Broadcast Received bits shall be reset to the released state. Resetting the User Defined bit in the SLAVE Status Register shall be optional.

5.3.4 CONTEND for Bus. This command shall cause all SLAVEs with a bit requiring a SLAVE interrupt asserted within the SLAVE Status Word, to CONTEND for the bus as described in Section "5.2.9 TM-Bus Contention" on page 21.

5.3.5 Enable Interrupt. When this command is received, the SLAVE shall be allowed to interrupt during IDLE or PAUSE states. The Interrupt Enable bit shall be set in the SLAVE status register.

5.3.6 Disable Interrupt. When this command is received, the SLAVE interrupt capability shall be disabled. The Interrupt Enable bit in the SLAVE status register shall be reset to a released state.

5.3.7 Multicast Select 0. When this command is received, the SLAVE shall be placed in multicast group 0 and the SLAVE Status Register Multicast Select bits shall be set to '00'. This shall enable the SLAVE to respond to command headers with an address field equal to ('FC' HEX).

5.3.8 Multicast Select 1. When this command is received, the SLAVE shall be placed in multicast group 1 and the SLAVE Status Register Multicast Select bits shall be set to '01'. This shall enable the SLAVE to respond to command headers with an address field equal to ('FD' HEX).

5.3.9 Multicast Select 2. When this command is received, the SLAVE shall be placed in multicast group 2 and the SLAVE Status Register Multicast Select bits shall be set to '10'. This shall enable the SLAVE to respond to command headers with an address field equal to ('FE' HEX).

5.3.10 Multicast Select 3. When this command is received, the SLAVE shall be placed in multicast group 3 and the SLAVE Status Register Multicast Select bits shall be set to '11'. This shall enable the SLAVE to respond to command headers with an address field equal to ('FF' HEX).

Command Field (MSB) (8)	(LSB) (2)	Command
0000000		READ STATUS
0000001		INITIALIZE MODULE
0000010		RESET SLAVE
0000011		CONTEND FOR BUS
0000100		MULTICAST SELECT 0
0000101		MULTICAST SELECT 1
0000110		MULTICAST SELECT 2
0000111		MULTICAST SELECT 3
0001000		ENABLE INTERRUPT
0001001		DISABLE INTERRUPT
0001010		RESERVED
0001011		RESERVED
0001100		RESERVED
0001101		RESERVED
0001110		RESERVED
0001111		RESERVED

Figure 16. Standard Commands

5.4 TM-Bus Synchronization/Initialization. The bus shall be initialized when both the MASTER DATA and CONTROL lines are simultaneously released, forcing the bus into the IDLE state. All SLAVEs on the bus shall then be capable of transactions over the bus. If desired, the command 'Reset SLAVE' may then be broadcast to bring all bus lines and SLAVEs to an error-free quiescent state. When a SLAVE receives the Reset command, all of the SLAVE Status Register bits are reset, its multicast select mode will be reset to '00', and SLAVE interrupts will be disabled.

A broadcast or multicast reset command shall set the Broadcast/Multicast Received bit.

5.5 TM-Bus Mastership. The TM-Bus shall have single MASTER operations. This specification shall not preclude the ability for systems to have more than one MASTER and a method to switch mastership of the bus independent of the four signal lines defined in this specification.

5.6 TM-Bus Error Handling. If a parity error is detected in a HEADER packet, the SLAVE shall not execute the command and shall set the Bus Error bit in the SLAVE status register and signal an interrupt as described in Section "5.2.8 TM-Bus Interrupts" on page 20. When a parity error is detected by the addressed SLAVE while receiving DATA packets, the SLAVE shall set the Bus Error bit in the SLAVE's Status Register and signal an interrupt as described in Section "5.2.8 TM-Bus Interrupts" on page 20.

Stuck-at-0 bus conditions are detected by the odd packet parity scheme as described in Section "5.2.2.2 Packet Parity" on page 11. Stuck-at-1 bus conditions shall be detected as an illegal command as described in Section "5.3 Command Definitions" on page 24.

To insure error free reception during broadcast or multicast, the read status register command should be broadcast or multicast first to clear each SLAVE's broadcast/multicast received bit. After the data or command is broadcasted or multicasted, the broadcast/multicast received bit should be checked by reading the status register of each SLAVE one at a time.

5.7 TM-Bus Testing. On-line testing of the bus is performed as a result of its normal operation. Off-line or power-up testing may be accomplished through the use of bus exercise routines and bus wrap/hand-shaking tests. The MASTER shall be able to send bad parity or set any SLAVE's Event Occurrence bit and check for proper SLAVE response utilizing user definable commands for test flexibility.

6 NOTES

Any comments should be submitted to:

J.P. Letellier
Space and Warfare Systems Command
Code 614
Washington, D.C. 20363-5100
phone (202) 692-0701

APPENDIX I

10 GLOSSARY

EOM	-	End of Message
Hz	-	Hertz
LSB	-	Least Significant Bit
MHz	-	Megahertz, 1 million cycles per second
mA	-	Milliamperes, 1 thousandth of an Ampere
MID	-	Module Identification
MIP	-	Module Identification Parity
MSB	-	Most Significant Bit
nh	-	Nanohenry, 1 billionth of a Henry
ns	-	Nanosecond, 1 billionth of a Second
pF	-	Picofarad, 1 trillionth of a Farad
TBD	-	To Be Defined
TM-Bus	-	Test and Maintenance Bus
uA	-	micro Amperes, 1 millionth of an Ampere

DRAFT

SPA 90099001A
16 January 1987

HIGH SPEED DATA BUS

SYSTEM SPECIFICATION
CDRL SEQUENCE NO. 10

PRELIMINARY

PREPARED BY

COLLINS GOVERNMENT AVIONICS DIVISION

ROCKWELL INTERNATIONAL

31 OCTOBER 1986

PREPARED FOR

PAVE PILLAR PROGRAM OFFICE

AFWAL/AAAS-3

WRIGHT PATTERSON AIR FORCE BASE

OHIO

PREPARED BY: M. B. Modrow
Technical Manager

APPROVED BY: J. N. Senko
Program Manager

CONTRACT NO. F33615-83-C-1036

UNCLASSIFIED

D

DRAFT

TABLE OF CONTENTS PAGE

1.	SCOPE	8
2.	APPLICABLE DOCUMENTS	8
2.1	Government Documents	8
2.2	Non-Government Documents	8
3.	REQUIREMENTS	8
3.1	System Definition	8
3.1.1	General Description	8
3.1.1.1	Conventions	10
3.1.2	Mission	10
3.1.3	Threat	10
3.1.4	System Diagrams	10
3.1.4.1	Coaxial Media Topology	10
3.1.4.2	Fiber Optic Media Topology	10
3.1.5	Interface Definition	12
3.1.5.1	Terminal/User Interface	12
3.1.5.1.1	Transmission Class of Service	12
3.1.5.1.1.1	Data Transmit Transaction Sequence	13
3.1.5.1.1.2	Transmit Block Size	13
3.1.5.1.2	Receive Operation	13
3.1.5.1.2.1	Data Receive Transaction Sequence	13
3.1.5.1.3	Status Request Transaction Sequence	14
3.1.5.2	Coaxial Terminal/Media Interface	14
3.1.5.2.1	Tx Interface	14
3.1.5.2.2	Rx Interface	14
3.1.5.2.3	Tx Switch Interface	15
3.1.5.2.4	Bus Mainline Interface	15
3.1.5.2.5	Waveform	15
3.1.5.2.5.1	Receiver Waveform	15
3.1.5.2.5.1.1	Amplitude Change	15
3.1.5.2.5.1.2	Maximum Signal	15
3.1.5.2.5.1.3	Timing	17
3.1.5.2.5.2	Transmitter Waveform	17
3.1.5.2.5.2.1	Amplitude	17
3.1.5.2.5.2.2	Ripple and Droop	17
3.1.5.2.5.2.3	Timing	17
3.1.5.2.5.2.4	Risetime/Falltime	17
3.1.5.3	FO Terminal/Media Interface	17
3.1.5.3.1	Tx Interface	17
3.1.5.3.2	Rx Interface	18
3.1.5.3.3	Waveform	18
3.1.5.3.3.1	Risetime	18
3.1.5.3.3.2	Falltime	18
3.1.5.3.3.3	Ripple/Droop	18
3.1.5.3.3.4	On/Off Ratio	18
3.1.5.3.3.5	Symmetry	18
3.1.6	Government Furnish Property List	18
3.1.7	Operational and Organizational Concepts	18
3.2	Characteristics	18
3.2.1	Performance Characteristics	20

TABLE OF CONTENTS (Cont'd) PAGE

3.2.1.1 Message Format 20
3.2.1.1.1 Preamble 22
3.2.1.1.2 Start Delimiter 22
3.2.1.1.3 Frame Control 22
3.2.1.1.3.1 Token Management Message. 22
3.2.1.1.3.2 Data Message. 23
3.2.1.1.3.3 Maintenance Message 23
3.2.1.1.3.3.1 Loopback Message. 24
3.2.1.1.3.3.2 Set Topology Memory 24
3.2.1.1.3.3.3 Topology Message. 25
3.2.1.1.3.3.4 Set Parameter 25
3.2.1.1.3.3.5 Report Parameter. 26
3.2.1.1.3.3.6 Parameter Message 26
3.2.1.1.3.3.7 Report Terminal Statistics. 26
3.2.1.1.3.3.8 Statistics Report 27
3.2.1.1.3.3.9 Set Message Filter. 28
3.2.1.1.3.3.10 Report Message Filter 28
3.2.1.1.3.3.11 Message Filter Report 28
3.2.1.1.3.3.12 Set Realtime Clock. 28
3.2.1.1.3.3.13 Realtime Clock Report 28
3.2.1.1.3.4 Alternate Control Message 29
3.2.1.1.4 Source Address. 29
3.2.1.1.5 Destination Address 29
3.2.1.1.5.1 Token Message 29
3.2.1.1.5.2 Maintenance Message 29
3.2.1.1.5.3 Data Message. 29
3.2.1.1.5.3.1 Singlecast Address. 29
3.2.1.1.5.3.2 Content Address 30
3.2.1.1.5.3.3 Broadcast Address 30
3.2.1.1.6 Word Count. 30
3.2.1.1.7 Data. 30
3.2.1.1.8 Frame Check Sequence. 30
3.2.1.1.9 End Delimiter 30
3.2.1.1.10 Maximum Transmission Time 31
3.2.1.1.11 Intermessage Gap. 31
3.2.1.1.11.1 Terminal Response Time. 31
3.2.1.1.11.2 Receiver Recovery 31
3.2.1.1.12 Redundancy. 31
3.2.1.2 MED Layer Performance 31
3.2.1.2.1 Coaxial Media Unique Characteristics. 31
3.2.1.2.2 FO Media Unique Characteristics 32
3.2.1.3 PHY Layer Performance 32
3.2.1.3.1 Coaxial TRU 33
3.2.1.3.2 FO TRU. 33
3.2.1.4 MAC Layer Performance 35
3.2.1.4.1 Transaction Data Unit Descriptions. 37
3.2.1.4.1.1 Tx Unit 37
3.2.1.4.1.2 Terminal Management Unit. 37
3.2.1.4.1.3 Tx Data Unit. 37
3.2.1.4.1.4 Tx Control Unit 37
3.2.1.4.1.5 Terminal Control Unit 38
3.2.1.4.1.6 Tx Message. 38

TABLE OF CONTENTS (Cont'd) PAGE

3.2.1.4.1.7 Rx Message 38
3.2.1.4.1.8 Rx Data Unit 38
3.2.1.4.1.9 Rx Control Unit 38
3.2.1.4.1.10 Rx Unit 39
3.2.1.4.1.11 Configuration Reporting Unit 39
3.2.1.4.1.12 Terminal Status Unit 39
3.2.1.4.2 Functional Requirements 39
3.2.1.4.2.1 Steady State Token Passing Operation 39
3.2.1.4.2.2 Deleted
3.2.1.4.2.3 Realtime Topology Adaptation 41
3.2.1.4.2.4 Initialization Process 41
3.2.1.4.2.5 Priority 41
3.2.1.4.2.6 Maintenance Functions 41
3.2.1.4.2.7 Realtime Clock 42
3.2.1.4.2.8 Acknowledgment 42
3.2.1.4.2.9 Retry 42
3.2.1.4.2.10 Redundant Bus Operation 43
3.2.1.4.2.11 Message Filter 43
3.2.1.4.2.12 Topology Memory 43
3.2.1.4.2.13 Network Activity Monitor 44
3.2.1.4.3 Timers 44
3.2.1.4.3.1 Token Hold Timer 44
3.2.1.4.3.2 Token Rotation Timers 46
3.2.1.4.3.3 Ring Maintenance Timer 46
3.2.1.4.3.4 Solicit Entry Timer 46
3.2.1.4.3.5 Network Inactivity Timer 47
3.2.1.4.3.6 Response Window Timer 47
3.2.1.4.3.7 Retry Limit Timer 47
3.2.1.4.3.8 Realtime Clock Synchronization Timer 47
3.2.1.4.4 State Machine Descriptions 48
3.2.1.4.4.1 Access and Protocol Machine 48
3.2.1.4.4.1.1 Off-Line State 48
3.2.1.4.4.1.2 Idle State 48
3.2.1.4.4.1.3 USE_TOKEN State 48
3.2.1.4.4.1.4 PASS_TOKEN State 52
3.2.1.4.4.1.5 ENTER_NET State 52
3.2.1.4.4.1.6 CLAIM_TOKEN State 52
3.2.1.4.4.1.7 SOLICIT ENTRY Process 55
3.2.1.4.4.1.8 LEAVE_NET Process 57
3.2.1.4.4.1.9 COLD_START Process 57
3.2.1.4.4.1.10 RESTART Process 59
3.2.1.4.4.1.11 Realtime Clock Management Process 59
3.2.1.4.4.1.12 Redundant Bus Protocol 61
3.2.1.4.4.2 User Interface Machine Description 62
3.2.1.4.4.3 Receive Machine Description 63
3.2.1.4.4.4 Transmit Machine Description 63
3.2.1.5 Error Recognition and Recovery 63
3.2.1.5.1 Recognition Strategies 67
3.2.1.5.1.1 Receiver Data Error 67
3.2.1.5.1.2 Token Lost Error 67
3.2.1.5.1.3 Receiver Sync Error 67
3.2.1.5.1.4 Receiver Framing Error 67

TABLE OF CONTENTS (Cont'd) PAGE

3.2.1.5.1.5	Word Count Error	67
3.2.1.5.1.6	Deleted	
3.2.1.5.1.7	Bus Activity Error	67
3.2.1.5.1.8	Token Hold Error	68
3.2.1.5.1.9	Network Initialize Error	68
3.2.1.5.1.10	Transmitter Error	68
3.2.1.5.1.11	Receiver Error	68
3.2.1.5.1.12	Token Error	68
3.2.1.5.2	Recovery Strategies	69
3.2.1.5.2.1	Alternate Bus Procedure	69
3.2.1.5.2.2	Retry Procedure	69
3.2.1.5.2.3	Recover Token Procedure	70
3.2.1.5.2.4	Restart	71
3.2.2	Physical Characteristics	71
3.2.3	Reliability	71
3.2.4	Maintainability	71
3.2.5	Availability	71
3.2.6	System Effectiveness Models	71
3.2.7	Environmental Conditions	71
3.2.8	Nuclear Control Requirements	71
3.2.9	Transportability	71
3.3	Design and Construction	71
3.3.1	Materials, Processes, and Parts	71
3.3.2	Electronmagnetic Radiation	71
3.3.3	Nameplates and Product Markings	72
3.3.4	Workmanship	72
3.3.5	Interchangeability	72
3.3.6	Safety	72
3.3.7	Human Performance/Human Engineering	72
3.4	Documentation	72
3.5	Logistic	72
3.6	Personnel and Training	72
3.7	Functional Area Characteristics	72
3.7.1	Coaxial Bus Functional Descriptions	72
3.7.1.1	Coax Bus Terminal	72
3.7.1.2	Coupler	74
3.7.1.3	Mainbus	74
3.7.1.4	Stub Bus	74
3.7.1.5	Termination	74
3.7.2	FO Bus Function Descriptions	74
3.7.2.1	FO Bus Terminal	74
3.7.2.2	Star Coupler	74
3.7.2.3	Fiber Stub	75
3.7.2.4	Attenuator	75
3.8	Precedence	75
4.	QUALITY ASSURANCE PROVISIONS	75
5.	PREPARATION FOR DELIVERY	75
6.	NOTES	75
10.	GLOSSARY OF ACRONYMS	76
20.	PI-BUS INTERFACE DESIGN GUIDE	78

LIST OF FIGURES

Figure	Title	Page
1	HSDB Network - General Topology	9
2	HSDB Network Coaxial/FO Topology	11
3	HSDB Coaxial MEDIA Waveform	16
4	HSDB FO Media Waveform	19
5	HSDB Functional Partitioning	21
6	PHY Layer Functional Diagram	34
7	MAC Layer Functional Description	36
8	Token Passing Mode Information Transfer Format	40
9	Terminal Timer Relationships	45
10	APM Simplified State Diagram	49
11	USE_TOKEN Control Flow Diagram	51
12	PASS_TOKEN Control Flow Diagram	53
13	ENTER_NET Control Flow Diagram	54
14	CLAIM_TOKEN Control Flow Diagram	54
15	SOLICIT ENTRY Control Flow Diagram	56
16	COLD_START Control Flow Diagram	58
17	Realtime Clock Management Control Flow Diagram	60
18	Receive Machine Functional Block Diagram	64
19	Transmit Machine Functional Block Diagram	65
20	HSDB System Functional Partitioning	73

LIST OF TABLES

Table	Title	Page
I	Transmit Operation	13
II	Receive Operation	14
III	Status Reporting Operation	14
IV	Bus Message Composition	20
V	Start Delimiter	22
VI	Token Management Frame FC Construct	22
VII	Data FC Construct - Class of Response	23
VIII	Data FC Construct - Priority	23
IX	Maintenance FC Construct	24
X	Parameter Descriptors	25
XI	Statistic Descriptors	26
XII	TMU/TCU Functions	37
XIII	TXC Functions	38
XIV	CRU Functions	39
XV	Topology Adaptation Transaction Sequence	41
XVII	Error Recognition and Recovery	66

PREFACE

Following publication of this specification, a High Speed Data Bus Critical Design Review of Protocol Definition was held at Wright-Patterson Air Force Base on 13 January 1987. The following list of planned changes to the PAVE PILLAR High Speed Data Bus System Specification have resulted due to that review and later interface meetings with the VHSIC 1750A Computer contractors. These planned changes also reflect an attempt to bring the specification closer to that of the SAE AE-9B Linear Token Passing Bus. A revised High Speed Data Bus System Specification incorporating these changes will be available in March 1987.

1. INITIALIZATION - Method has been changed to the new method proposed by Lockheed-Georgia. This is necessary to reduce the chance of more than one terminal attempting to take control of the bus.

2. MEDIA/COUPLER - Specification of media characteristics (e.g., fiber size) will be relocated to appendices (one each 100um, 200um, coax). Minimum and maximum loss will be specified.

3. PREAMBLE - Sixteen (16) bits. Not sent between concatenated messages.

4. MESSAGE/FRAME FORMATS - Changes will be made to agree more closely with SAE. Unique types will be defined but not required.

5. TIMERS - Exhaustive message class deleted. Section will be rewritten for better understanding.

6. RETRIES OF TOKEN PASS - Change to choice of 0 or 1, default 1.

7. ERROR STATUS/STATISTICS - Change to follow SAE set as required, others as defined by 31 October 86 specification will remain defined, but will be optional.

8. REALTIME CLOCK - Change name to "Reference Timer." Reduce accuracy requirement to 10E-4 (.01%), increase update rate to 50mS, eliminate drift rate compensation requirement.

9. TERMINOLOGY - An attempt will be made to agree with the terminology used by the SAE.

10. LOOPBACK MESSAGE - The Frame Control (FC) field will be changed to indicate "This is a loopback message" to prevent infinite loopback.

11. TOPOLOGY MAP - This proposed change would simplify the implementation by reducing the memory required.

HIGH SPEED BUS SYSTEM SPECIFICATION

1. SCOPE

This specification defines performance, design, and development requirements for a High Speed Data Bus (HSDB) data communications system. The HSDB shall provide a real time communications network of a maximum of 64 terminals with addressing growth to allow future expansion to 127 terminals. Terminals may be separated by a maximum of three hundred feet.

2. APPLICABLE DOCUMENTS

The following documents of the issue in effect on 26 September 1983, or as noted below, form a part of this specification to the extent specified herein.

2.1 Government documents

MIL-E-5400T	Electronic Equipment, Aerospace, General Specification For
DOD-D-1000B	Drawings, Engineering and Associated Lists, Military Specifications For

2.2 Non-government documents

a. IEEE-STD-802 Local Area Networks

3. REQUIREMENTS

3.1 System Definition

The HSDB system shall provide high quality data communications among a network of user terminals at a bit rate of 50 Mbps. The HSDB shall support two forms of media, coaxial cable or fiber optic. A form of token passing protocol shall normally govern access to the network and coordination of data flow.

3.1.1 General Description

The HSDB network shall consist of a set of user terminals connected to a common media as illustrated by figure 1. The media shall be either coaxial cable (wire) or fiber optic (FO) for any specific network. Between two and sixty-four user terminals shall connect to the media using stub interconnect of the type compatible with the media. Each user terminal shall also connect to one or more users. Each terminal shall perform the functions of the Media Access Control (MAC) layer and the Physical (PHY) layer as described in IEEE-STD-802. This shall allow the network to appear transparent to the user process.

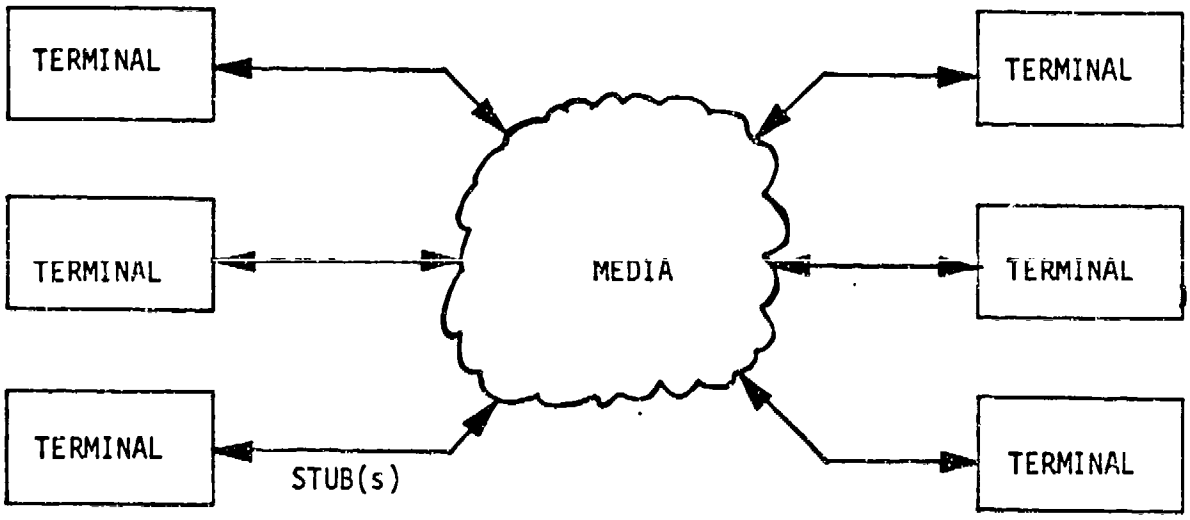


FIGURE 1

HSDB NETWORK - GENERAL TOPOLOGY

A form of token passing protocol shall be used to perform the management functions of the network. The token controls the right of access to the media; the terminal which holds (possesses) the token has momentary singular control over the network. The token is passed from terminal to terminal in a deterministic fashion forming a logical ring. Maintenance functions within each terminal provide for initialization of the logical ring, lost token recovery, addition of stations while the network is in operation, and general network monitoring.

3.1.1.1 Conventions

Several conventions are followed throughout this document. These are described below:

a. Numbering of Bits: Digital data fields are labeled using the convention of bit #0 as most significant bit and bit #n as least significant bit.

b. Transmission Sequence: Lower numbered bits to be transmitted on the network are sent first.

3.1.2 Mission

The HSDB shall be a general purpose digital communication network and shall be usable in ground based, shipboard, and aircraft applications within the environmental and performance constraints described herein.

3.1.3 Threat (not applicable)

3.1.4 System Diagrams

HSDB systems shall be implemented using a single media form, either coaxial cable of FO.

3.1.4.1 Coaxial Media Topology

Networks implemented using coaxial cable media shall use a linear bus topology as illustrated by figure 2a. The bus shall be loaded using a resistive termination at each extremity. A bi-directional bus coupler shall be used to tap the bus for each terminal. Dedicated transmit (TX) and Receiver (RX) coaxial stubs shall connect between the terminal and the coupler. A TX/RX turnaround stub between the terminal and the coupler shall decouple the transmit port from the bus unless the terminal is transmitting. Signal amplitude and loss values are shown for a 64 terminal network.

3.1.4.2 Fiber Optic Media Topology

Networks implemented using FO media shall use a passive star topology as illustrated by figure 2b. A single centralized star coupler shall provide transmit and receive access to each terminal. Dedicated TX and RX FO stubs shall connect between the terminal and the coupler. Signal amplitude and loss values are shown for a 64 terminal network.

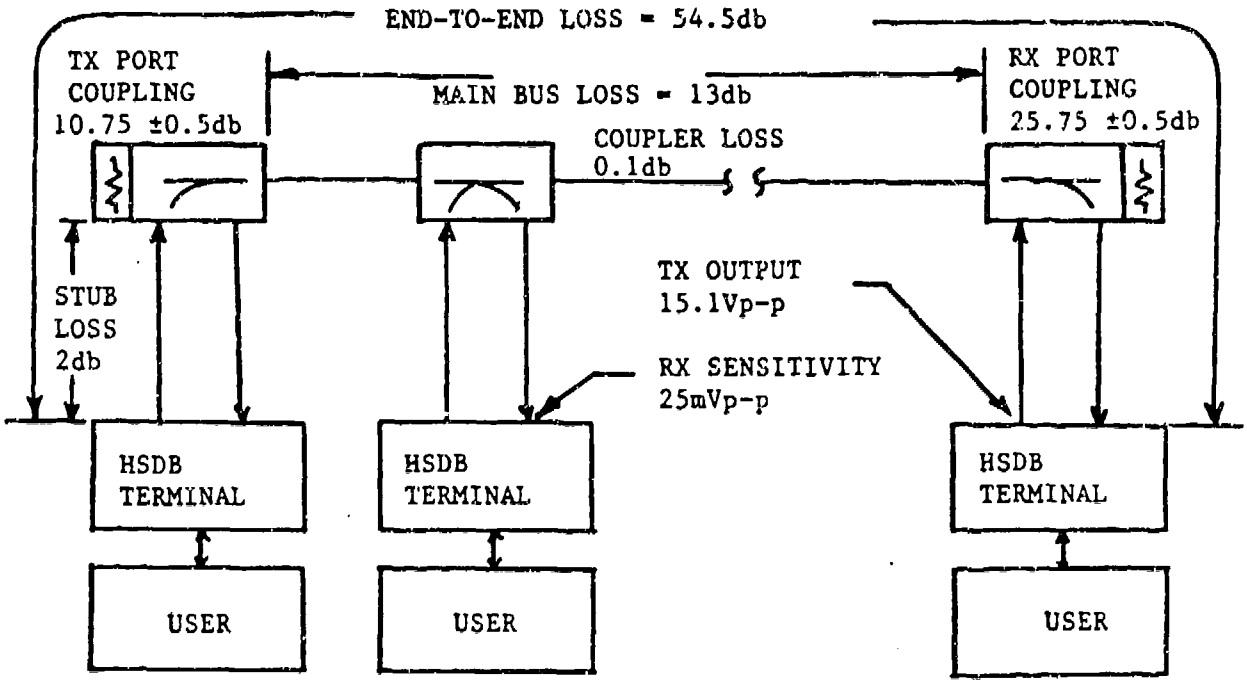


FIGURE 2a
COAXIAL NETWORK POWER BUDGET

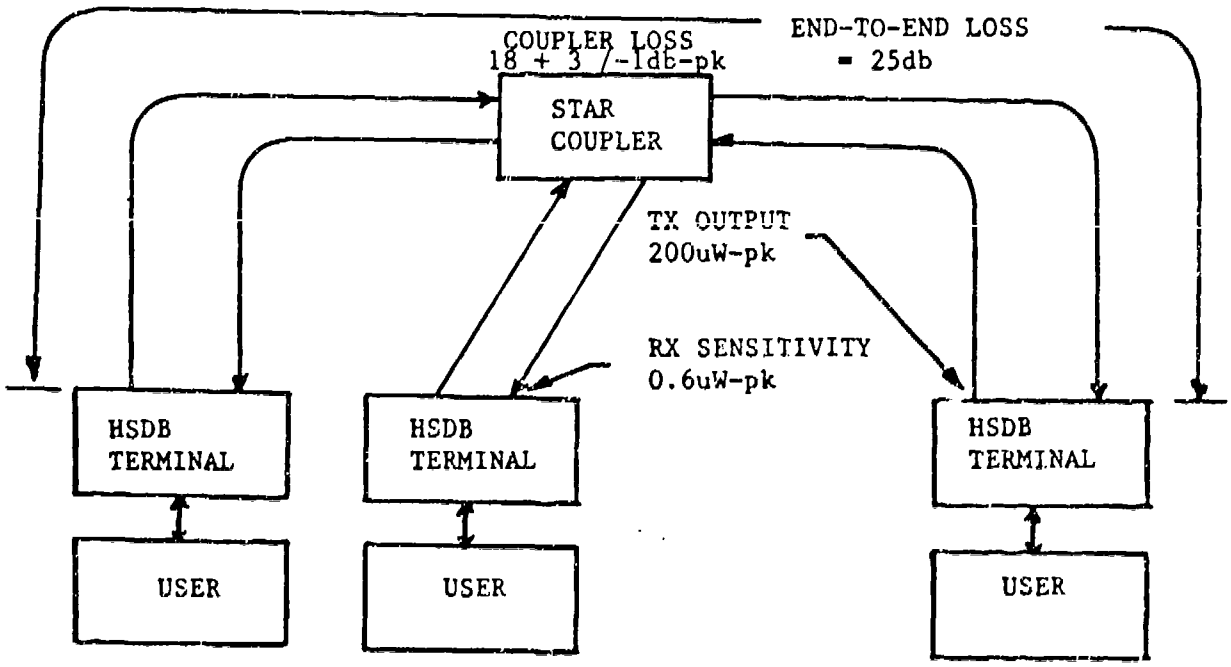


FIGURE 2b
FIBER OPTIC NETWORK POWER BUDGET

3.1.5 Interface Definition

The interface between the terminal and the user process shall be identical for both coaxial and FO implementations of the HSDB. The interface between the terminal and the media layer and within the media layer, shall be implementation unique.

3.1.5.1 Terminal/User Interface

The functional interface between the terminal and the user shall consist of a set of 4 data transaction units.

a) TX Unit - The TX Unit (TXU) shall be comprised of data to be sent from the terminal to one or more other terminals in the network.

b) RX Unit - The RX Unit (RXU) shall consist of data received by the terminal which is addressed to itself, and which contains no detected errors.

c) Terminal Management Unit - The Terminal Management Unit (TMU) shall consist of instructions from the user to the terminal to set the operating mode of the terminal, request transmission of data, or request network/terminal maintenance information.

d) Terminal Status Unit - The Terminal Status Unit (TSU) shall consist of information from the terminal to the user to indicate receipt of a message from the network or to respond to a request for terminal or network status information.

The electrical, physical, and format description of this interface is beyond the scope of this document. The information content shall, however, be consistent with the requirements of this document.

3.1.5.1.1 Transmission Class of Service

The terminal shall support two classes of service as defined below. Terminals shall operate using exhaustive class of service unless priority class is enabled following initialization of the network.

a. Exhaustive: The terminal shall transmit messages to maximally utilize the token hold time each time the token is held, within the constraints of the token hold timer. All messages shall be scheduled at P0 level independent of the priority level under which they were placed in the queue. The scheduling algorithm shall be first-in-first-out (FIFO).

b. Multiple level priority: The terminal shall transmit all messages held in queue within the constraints of the token rotation timers and the token hold timer. The

scheduling algorithm shall be FIFO within each priority level.

3.1.5.1.1.1 Data Transmit Transaction Sequence

The sequence of transactions between the user and the terminal for initiation of a data transmit operation shall be as described in Table I.

TABLE I
TRANSMIT OPERATION

SEQ	TRANSACTION USER	TERM	DESCRIPTION
1.	TMU	--->	:Request to send, block size, class of service, destination, subaddress
2.	<---	TSU	:Buffer available/busy, destination off-line
3.	TXU	--->	:Data
4.	TMU	--->	:Confirm buffer filled, abort
5.	<---	TSU	:Acknowledge data sent (and ack if requested)

A compatible methodology for PI-Bus interface is described in section 2.0.

3.1.5.1.1.2 Transmit Block Size

The terminal shall contain sufficient local storage to accept a single message of 4096 words from the user during a single transfer. Multiple smaller messages may share the transmit block at the discretion of the user. Each message shall be characterized by its unique class of service.

3.1.5.1.2 Receive Operation

Data received from the bus shall be transferred to the user at a rate controlled by the user. The terminal shall be capable of buffering any number of consecutive messages from the bus within the constraints of the terminal buffer size. As a minimum, the buffer shall accommodate a single message of 4096 words or multiple messages of that equivalent size. If all receive buffers are full when a new message is received, the new message shall be dropped and the terminal shall continue normal operation.

3.1.5.1.2.1 Data Receive Transaction Sequence

The sequence of transactions required between terminal and user for initiation of a data received operation shall be as described in Table II.

TABLE II
RECEIVE OPERATION

SEQ	TRANSACTION USER	TRANSACTION TERM	DESCRIPTION
1.	<---	TSU	:Data block size, location of buffer, source, class of service
2.	RXU	<---	:Data
3.	TMU	--->	:Confirmation (free buffer), location of buffer

3.1.5.1.3 Status Request Transaction Sequence

The sequence of transactions required between terminal and user for initiation of a status reporting operation shall be as described in Table III.

TABLE III
STATUS REPORTING OPERATION

SEQ	USER	TERM	DESCRIPTION
1.	TMU	--->	:Request for status
2.	<---	TSU	:Status information

3.1.5.2 Coaxial Terminal/Media Interface

The interface between a coaxial type user terminal and the media shall consist of four coaxial interfaces, a TX interface, a RX interface, a TX switch interface, and bus mainline interface. In the case of terminals with dual redundant HSDB bus ports, each port shall be independent and shall meet the requirements stated. In the case of single port terminals, the "A" bus shall be the active port and the "B" bus shall not be implemented.

3.1.5.2.1 TX Interface

The TX interface shall provide the means by which messages are placed on the bus from the terminal.

a. Connector Type - As defined by applicable critical item specification
b. Bit Rate - 50Mbps $\pm$ 12.5 KHz
c. Modulation Type - Manchester II
d. Signal Level - 15V $\pm$ 1Vp-p into 50 ohms for 50 MHz component

3.1.5.2.2 RX Interface

The RX interface shall provide the means by which messages are recovered from the bus by the terminal.

a. Connector Type - As defined by applicable critical item specification

b.	Bit Rate	-	50Mbps ±25KHz
c.	Modulation Type	-	Manchester II
d.	Minimum Signal	-	0.025Vp-p (50MHz component)
e.	Maximum Signal	-	0.270Vp-p (50MHz component)
f.	Impedance	-	50 ohms ± 2 ohms

3.1.5.2.3 TX Switch Interface

The TX switch interface shall provide the means by which the bus coupler disconnects the TX interface from the BUS mainline.

a.	Connector Type	-	As defined by applicable critical item specification
b.	Voltage output (TX)	-	11.0 ±4.0Vdc into 50 ohms
c.	Voltage output (RX)	-	-5.0 ±0.5V into open circuitry -2.5 ±0.2Vdc into 50 ohms
d.	Switching Time	-	TX state shall be asserted and stable within 100ns

3.1.5.2.4 Bus Mainline Interface

The bus mainline interface shall provide the means by which the bus interconnects between adjacent couplers.

a.	Connector Type	-	TNC (female polarity on coupler)
b.	Power Level	-	3.0Vp-p to 4.6Vp-p (50 MHz component)
c.	Impedance	-	50 ohms ± 1.25 ohms at 25MHz and 50MHz

3.1.5.2.5 Waveform

The critical characteristics of the waveform shall be the wave front detection period including the embedded state transition points and their spacing (in time). As shown in figure 3a, a wavefront detection period is defined within any 5ns segment during which:

a. The first 1ns period (zone A) exhibits a minimum rate of change of 2mv/ns.

b. The following 4ns period (zone B) exhibits the same polarity and is continuous.

3.1.5.2.5.1 Receiver Waveform

3.1.5.2.5.1.1 Amplitude Change

The amplitude of the waveform during Zone B of the wavefront detection period shall be between 10.5mv and 300mv.

3.1.5.2.5.1.2 Maximum Signal

The maximum signal, including overshoot and other waveform anomalies shall be not more than 500mv.

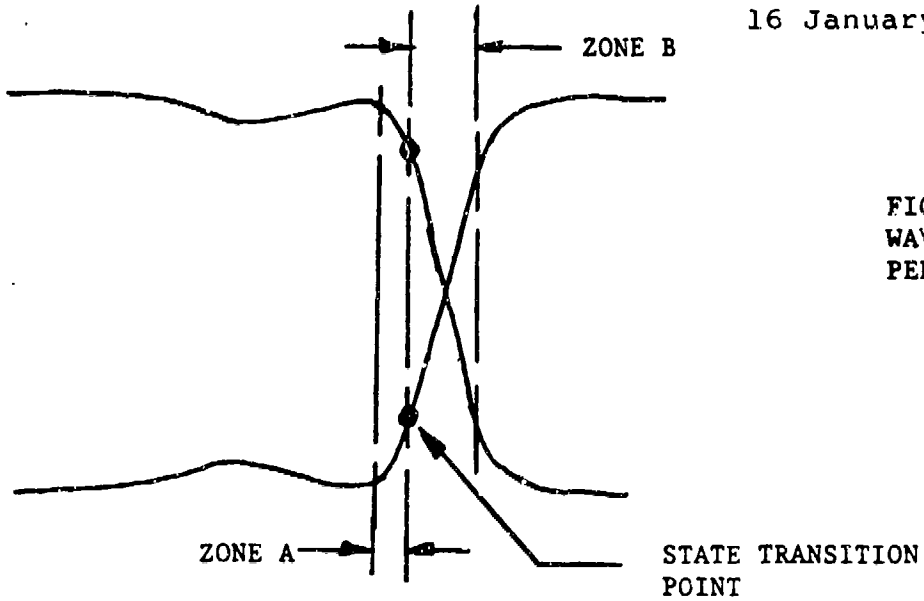


FIGURE 3a
WAVEFRONT DETECTION
PERIOD DEFINITION

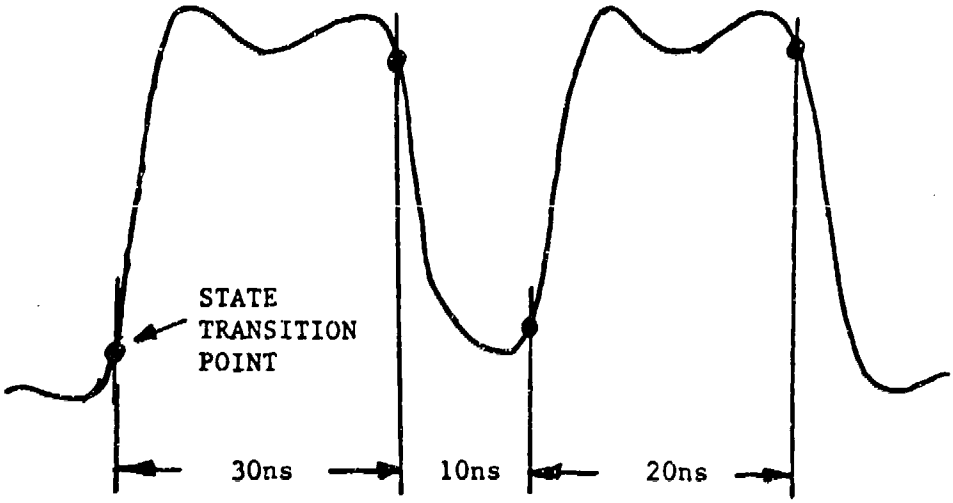


FIGURE 3b
RECEIVER INPUT
WAVEFORM

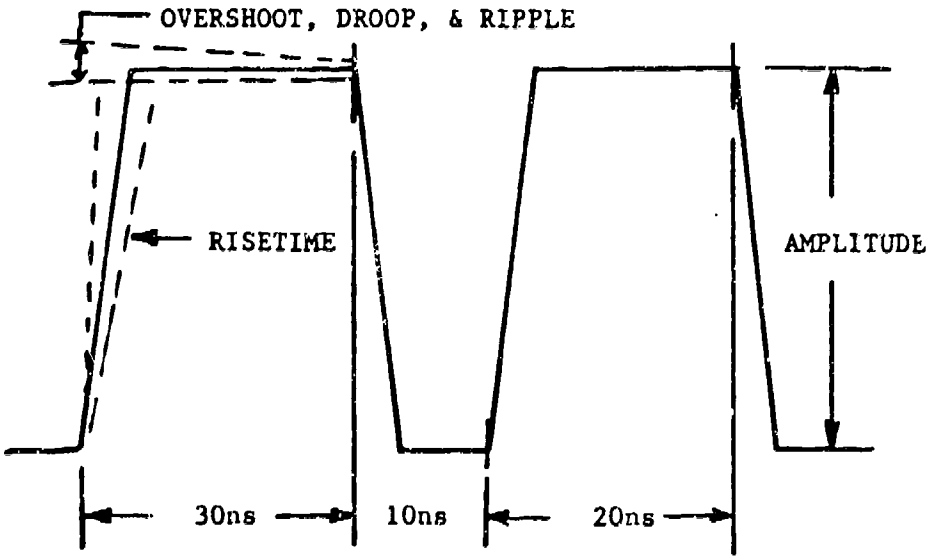


FIGURE 3c
TRANSMITTER OUPTUT
WAVEFORM

3.1.5.2.5.1.3 Timing

The period between adjacent state transition points of the waveform shall be either 10ns ±1ns or 20ns ±1ns for valid manchester logic symbols. The period between adjacent state transition points shall be 30ns ±1ns for the special symbols reserved for message delimiting (ref 3.2.1.1).

3.1.5.2.5.2 Transmitter Waveform

3.1.5.2.5.2.1 Amplitude

The peak-to-peak amplitude of the smoothed equivalent waveform shall be 15V ±1V into 50 ohms.

3.1.5.2.5.2.2 Ripple and Droop

The total ripple and droop component shall be less than 10% of the peak-to-peak amplitude of the smoothed equivalent waveform.

3.1.5.2.5.2.3 Timing

The period between adjacent state transition points shall be either 10ns ±1ns or 20ns ±1ns for valid manchester logic symbols. The period between adjacent state transition points shall be 30ns ±1ns for the special symbols reserved for message delimiting (ref 3.2.1.1).

3.1.5.2.5.2.4 Risetime/Falltime

The time between the 10% points and the 90% points (and inverse) shall be between 2ns and 6 ns.

3.1.5.3 FO Terminal/Media Interface

The interface between a FO type user terminal and the media shall consist of two discrete interfaces. In the case of terminals with dual redundant HSDE ports, each port shall be independent and shall meet the requirements stated. In the case of single port terminals, the "A" bus shall be the active port and the "B" bus shall not be implemented.

3.1.5.3.1 TX Interface

The TX interface shall be used to place properly modulated and formatted data packets on the bus under control of the user terminal.

a.	Connector Type	-	As defined by applicable critical item specification
b.	Wavelength	-	850 nm
c.	Power Lever (peak)	-	400uw-pk
d.	Modulation Type	-	Binary Manchester
e.	Fiber Type	-	100/140 micron, semi-graded index
f.	Numerical Aperture	-	0.29

3.1.5.3.2 RX Interface

The RX interface shall be used to recover properly modulated and formatted data packets from the bus.

a.	Connector Type	-	As defined by applicable critical item specification
b.	Wavelength	-	850 nm
c.	Modulation Type	-	Binary Manchester
d.	Fiber Type	-	100/140 Micron, semi-graded index
e.	Numerical Aperture	-	0.29

3.1.5.3.3 Waveform

The waveform as monitored at any transmitter or receiver port shall meet the following requirements (reference figure 4).

3.1.5.3.3.1 Risetime

Risetime of the optical pulse shall be less than 4 ns. No anomalies shall be present during the risetime portion of the pulse.

3.1.5.3.3.2 Falltime

Falltime of the optical pulse shall be less than 4 ns. No anomalies shall be present during the falltime portion of the pulse.

3.1.5.3.3.3 Ripple/Droop

The difference in peak amplitude between any two pulses in a message shall be less than 1.5db.

3.1.5.3.3.4 On/Off Ratio

The ratio between the 100% point of an optical pulse and the 0% point of segments of the waveform shall be no less than 45db.

3.1.5.3.3.5 Symmetry

The ratio of ontime to bit period shall be 0.5 $\pm$0.05 (for valid manchester symbols).

3.1.6 Government Furnished Property List (not applicable)

3.1.7 Operational and Organizational Concepts

The HSDB system shall be a general purpose 50Mbps digital data communications network and shall be usable in ground based, shipboard, and aircraft applications requiring 127 or fewer terminals within the environmental and performance envelope described herein.

3.2 Characteristics

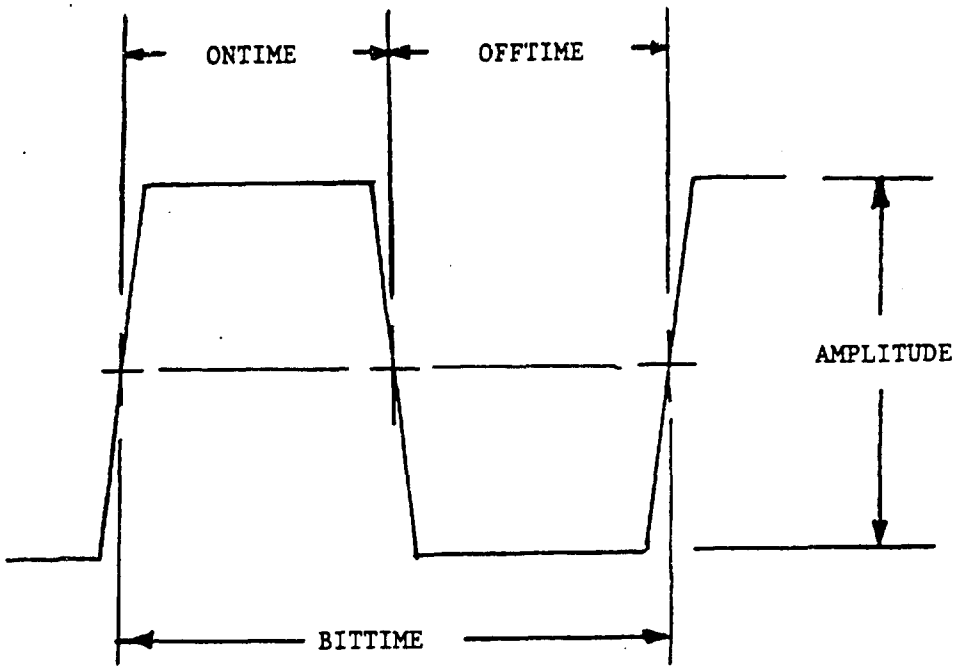


FIGURE 4a
FIBER OPTIC SINGLE BIT WAVEFORM

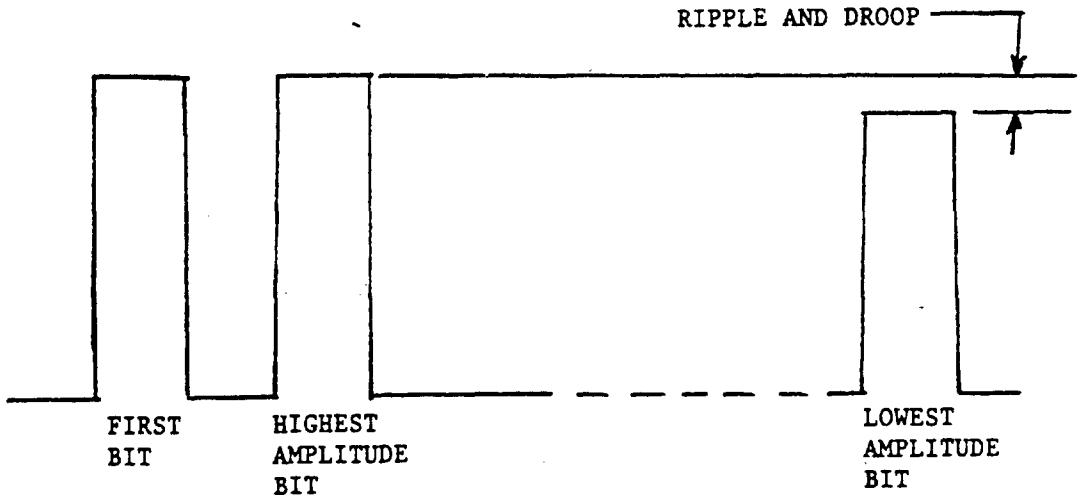


FIGURE 4b
FIBER OPTIC MESSAGE WAVEFORM

3.2.1 Performance Characteristics

The HSDB shall encompass the functions of the media layer (MED), physical layer (PHY) and Media Access Control (MAC) layer of a network as described in IEEE-STD-802. Figure 5 illustrates the relationship between IEEE-STD-802 functions and HSDB elements. The MED shall be implemented in the form of coaxial cable and couplers for coaxial type implementations and in the form of fiber optic cable and a coupler for FO type implementations. The PHY and the MAC shall be implemented in the form of electronic circuitry in the HSDB terminal. A unique PHY design shall be required for each type terminal, coaxial or FO. The MAC shall be identical for either implementation.

3.2.1.1 Message Format

Messages placed on the bus shall conform to the message format described below including the size and location of all transaction data unit fields and the hierarchy of functions within data units. Four message formats shall be supported by the HSDB, token messages, data messages, maintenance messages, and alternate control messages. The sequence of transaction data units comprising a data message is described in Table IV.

TABLE IV
BUS MESSAGE COMPOSITION

SEQUENCE	TRANSACTION DATA UNIT	SIZE
1	Preamble (PRE)	8 bits
2	Start Delimiter (SD)	4 bit times
3	Frame Control (FC)	8 bits
4	Destination Address (DA)	8 bits or 16 bits
5	Source Address (SA)	8 bits
6	Word Count (WC)	16 bits or 0 bits
7	Data	0 to 256 words
8	Frame Check Sequence (FCS)	16 bits
7a/8a	Additional data fields and FCS fields may be appended prior to ED for messages longer than 256 words	
9	End Delimiter (ED)	4 bit times

A valid abort sequence, PRE + SD +<---> + ED + ED, shall be defined. Any message length shall be allowed between the SD and ED fields, up to a maximum of 65,856 bits. The abort sequence shall be sent under certain conditions following detection of a transmitter error in order to allow receiving terminals to differentiate from network and receiver errors.

It shall be possible to concatenate messages within a single transaction. In this case, the messages shall be delimited by an ED + SD sequence without intervening PRE or intermessage gap.

The sequence of transaction data units comprising a token message shall be PRE + SD + FC + DA + SA + FCS + ED. The composition

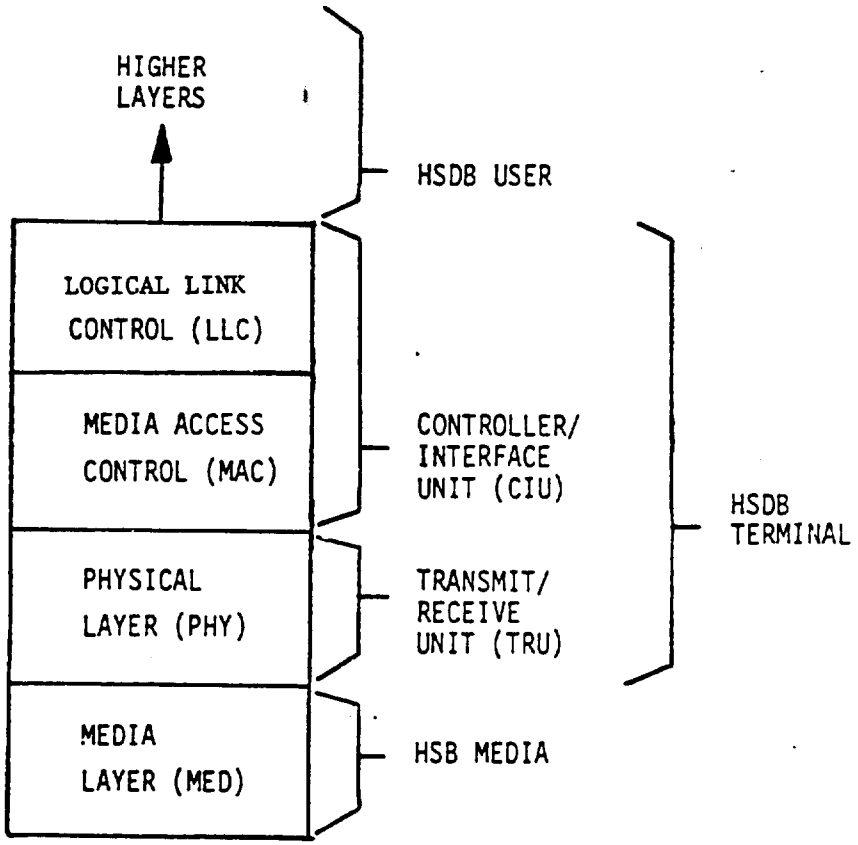


FIGURE 5

HSDB FUNCTIONAL PARTITIONING

of each field is described in Table IV. The DA for a token message shall always be 8 bits in length. The sequence of transaction data units comprising a data message shall be PRE + SD + FC + DA + SA + WC + data + FCS + ED. The DA for a data message shall always be 16 bits in length. The sequence of transaction data units comprising a maintenance message shall be PRE + SD + FC + DA + SA + <WC + data +> FCS + ED. The DA for a maintenance message shall always be 8 bits in length.

3.2.1.1.1 Preamble

The PRE field pattern shall precede each transmitted message. It shall consist of 8 bits of logic 1 symbols.

3.2.1.1.2 Start Delimiter

The SD field shall consist of the 4 bit time sequence of valid and invalid symbols as indicated in Table V.

TABLE V
START DELIMITER

Bit Time	Symbol
1	Valid logic "0"
2	Invalid symbol - no transition from bit #1
3	Invalid symbol - complement of bit #2
4	Valid logic "0"

3.2.1.1.3 Frame Control

The FC field shall be used to define the message type. The first two bits shall define the message format:

Bit #
0 1

00 - Token management message
01 - Data message
10 - Terminal maintenance message
11 - Alternate control message (reserved)

3.2.1.1.3.1 Token Management Message

Bits 2 through 7 of a token management message shall be encoded as defined by Table VI.

TABLE VI
TOKEN MANAGEMENT FC CONSTRUCT

Bit # 2 3 4 5 6 7	Definition	Message Construct
0 0 0 0 0 0	Illegal	
0 0 0 0 0 1	Solicit Entry	DA + SA + FCS
0 0 0 0 1 0	Set Successor	DA + SA + CS + FCS

TABLE VI (Cont'd)
TOKEN MANAGEMENT FC CONSTRUCT

Bit # 2 3 4 5 6 7	Definition	Message Construct
0 0 0 0 1 1	Pass Token	DA + SA + FCS
) 0 0 1 0 0	Set Predecessor	DA + SA + CP + FCS
0 0 0 1 0 1	Request Entry	DA + SA + FCS
0 0 0 1 1 1	Pass Token, Leave net	DA + SA + FCS
0 0 1 0 0 1	Solicit Reentry	DA + SA + FCS
0 1 0 1 0 1	Request Reentry	DA + SA + FCS
1 0 0 1 0 1	Claim Token	Paragraph 3.2.1.4.4.1.6
1 0 0 1 1 0 through 1 1 1 1 1 1	Reserved	

Where CS and CP are the 8-bit current successor and current predecessor addresses.

3.2.1.1.3.2 Data Message

Bits 2 through 7 of a data message shall specify requested class of response and priority as defined by Table VII and Table VIII.

TABLE VII
DATA FC CONSTRUCT - CLASS OF RESPONSE

Bit # 2 3 4	Class of Response	Message Construct
0 0 0	No acknowledgment requested	DA + SA + WC + Data + FCS
0 0 1	Reserved	
0 1 0	Reserved	
0 1 1	Reserved	

TABLE VIII
DATA FC CONSTRUCT - PRIORITY

Bit # 5 6 7	Priority
0 0 0	Highest Priority (P0)
0 0 1	Mid Priority (P1)
0 1 0	Low Priority (P2)
1 0 0	No Priority (P3)

3.2.1.1.3.3 Maintenance Message

Bits 2 through 7 of a maintenance message shall be encoded as defined by Table IX. Maintenance messages shall be scheduled in the transmit queue at the priority level described by Table IX unless explicitly scheduled at a different level by the user.

TABLE IX
MAINTENANCE FC CONSTRUCT

Bit # 2	3	4	5	6	7	Definition	Priority	Message Construct
0	0	0	0	0	0	RESET (off-line)	0	DA + SA + FCS
0	0	0	0	0	1	Set maintenance loopback	2	DA + SA + FCS
0	0	0	0	1	0	Disable Maintenance loopback	2	DA + SA + FCS
0	0	0	0	1	1	Loopback message	2	DA + SA + WC + Data + FCS
0	0	0	1	0	0	Set topology memory	0	DA + SA + WC + Data + FCS
0	0	0	1	0	1	Report topology memory	3	DA + SA + FCS
0	0	0	1	1	0	Topology message	3	DA + SA + WC + Data + FCS
0	0	0	1	1	1	Reserved		
0	0	1	0	0	0	Set parameter(s)	0	DA + SA + WC + Data + FCS
0	0	1	0	0	1	Report Parameter(s)	3	DA + SA + WC + Data + FCS
0	0	1	0	1	0	Parameter message	3	DA + SA + WC + Data + FCS
0	0	1	0	1	1	Reserved		
0	0	1	1	0	0	Reserved		
0	0	1	1	0	1	Report terminal statistic(s)	3	DA + SA + WC + Data + FCS
0	0	1	1	1	0	Statistic(s) report	2	DA + SA + WC + Data + FCS
0	0	1	1	1	1	Reserved		
0	1	0	0	0	0	Reserved		
0	1	0	0	0	1	Set Message Filter	0	DA + SA + WC + data + FCS
0	1	0	0	1	0	Report Message Filter	3	DA + SA + FCS
0	1	0	0	1	1	Message Filter Config. msg.	3	DA + SA + WC + data + FCS
0	1	0	1	0	0	Reserved		
0	1	0	1	0	1	Set Realtime Clock	0	DA + SA + WC + data + FCS
0	1	0	1	1	0	Report Realtime Clock	0	DA + SA + FCS
0	1	0	1	1	1	Realtime Clock Report	0	DA + SA + WC + data + FCS
0	1	1	0	0	0	Reserved		
0	1	1	0	0	1	Set Redundant Bus Mode	3	DA + SA + FCS
0	1	1	0	1	0	Reset Redundant Bus Mode	2	DA + SA + FCS

The content of maintenance messages shall conform to the construct defined in subsequent paragraphs. In all cases, the SA field shall contain the terminal address (TA) of the terminal originating the message and the DA field shall contain the terminal address of the terminal to which the message is directed.

3.2.1.1.3.3.1 Loopback Message

A terminal receiving a loopback message, when loopback mode is enabled, shall queue a complementary response message in which the SA and DA fields are interchanged. The data field shall be unchanged. The FCS shall be recalculated for the response message.

3.2.1.1.3.3.2 SET TOPOLOGY MEMORY

The data field of a SET_TOPOLOGY_MEMORY message shall consist of a sequence of 40 bit-records, one for each terminal included in the topology.

Bit(s)

0-7	-	Terminal address
8	-	Present - "1"/Absent - "0"
9	-	Online - "1"/Offline - "0"
10	-	Timekeeper - "1"/Not Timekeeper - "0"
11	-	Reserved (set to "0")
12	-	TX "A" Port Operational - "1"
13	-	TX "B" Port Operational - "1"
14	-	RX "A" Port Operational - "1"
15	-	RX "B" Port Operational - 1"
16-23	-	Predecessor Address (PA)
24-31	-	Successor Address (CS)
32-39	-	Reserved

Should the message contain an odd number of records, additional bits of logic 0 data shall be appended to the field sufficient to fill it to a length evenly divisible by 16. The WC field of a SET TOPOLOGY MEMORY message shall define the count of 16 bit frames which exist within the data field rather than the count of 40 bit records.

3.2.1.1.3.3.3 TOPOLOGY Message

The data field of a TOPOLOGY message shall be identical to that described for a SET TOPOLOGY MEMORY message.

3.2.1.1.3.3.4 SET PARAMETER

The data field of a SET PARAMETER message shall consist of a sequence of 24-bit records, 1 for each parameter to be set as defined by Table X. Unused bits shall be 0.

TABLE X
PARAMETER DESCRIPTORS

Bit 0 1 2 3 4 5 6 7	Parameter Bits 8 through 23
0 0 0 0 0 0 0 0	Reserved
0 0 0 0 0 0 0 1	Reserved
0 0 0 0 0 0 1 0	RWT value (16-23)
0 0 0 0 0 0 1 1	THT value (8-23)
0 0 0 0 0 1 0 0	TRT_P1 value (8-23)
0 0 0 0 0 1 0 1	TRT_P2 value (8-23)
0 0 0 0 0 1 1 0	TRT_P3 value (8-23)
0 0 0 0 0 1 1 1	RMT value (8-23)
0 0 0 0 1 0 0 0	SET value (8-23)
0 0 0 0 1 0 0 1	RLT value (20-23)
0 0 0 0 1 0 1 0	NIT value (16-23)
0 0 0 0 1 0 1 1 through 1 1 1 1 1 1 1 1	Reserved
	Reserved

Bits 8 through 23 shall contain the value to which the parameter is to be set. Should the message contain an odd number of parameter records, additional bits of logic 0 data shall be appended to the data field sufficient to fill it to a length evenly divisible by 16. The WC field of a set parameter message shall define the count of 16-bit frames which exist within the data field rather than the count of 24-bit records.

3.2.1.1.3.3.5 REPORT PARAMETER

The data field of a REPORT PARAMETER message shall consist of a sequence of 8-bit records, 1 for each parameter to be reported. Table X defines the relationship between parameter and data content with the following additions:

0000 1011 Reserved
0000 1111 Message Filter Size Description

Should the message contain an odd number of records, additional bits of logic 0 data shall be appended to the field sufficient to fill it to a length evenly divisible by 16. The WC field of a report parameter message shall define the count of 16-bit words which exist within the data field rather than the count of 8-bit records.

3.2.1.1.3.3.6 PARAMETER Message

The data ad WC fields of a PARAMETER message shall be identical to that described for a SET PARAMETER message with the following additions:

0000 1011 Reserved
0000 1111 Message Filter Size Description (20-23)

3.2.1.1.3.3.7 REPORT_TERMINAL_STATISTICS

The data field of a REPORT_TERMINAL_STATISTICS message shall consist of a sequence of 8 bit records, one for each statistic to be reported. Table XI defines the relationship between statistic and data content.

TABLE XI
STATISTIC DESCRIPTORS

Bit 0 1 2 3 4 5 6 7	Statistic
0 0 0 0 0 0 0 0	Terminal Status Summary
0 0 0 0 0 0 0 1	Number of P0 messages sent
0 0 0 0 0 0 1 0	Number of P0 messages received
0 0 0 0 0 0 1 1	Number of P1 messages sent
0 0 0 0 0 1 0 0	Number of P1 messages received
0 0 0 0 0 1 0 1	Number of P2 messages sent
0 0 0 0 0 1 1 0	Number of P2 messages received
0 0 0 0 0 1 1 1	Number of P3 messages sent
0 0 0 0 1 0 0 0	Number of P3 messages received
0 0 0 0 1 0 0 1	Number of tokens received

TABLE XI (Cont'd)
STATISTIC DESCRIPTORS

Bit 0	1	2	3	4	5	6	7	Statistic
0	0	0	0	1	0	1	0	Number of tokens viewed on net
0	0	0	0	1	0	1	1	Average token cycle time (usec)
0	0	0	0	1	1	0	0	Maximum token cycle time (usec)
0	0	0	0	1	1	0	1	Average transmission delay - P0 (usec)
0	0	0	0	1	1	1	0	Maximum transmission delay - P0 (usec)
0	0	0	0	1	1	1	1	Average transmission delay - P1 (usec)
0	0	0	1	0	0	0	0	Maximum transmission delay - P1 (usec)
0	0	0	1	0	0	0	1	Average transmission delay - P2 (usec)
0	0	0	1	0	0	1	0	Maximum transmission delay - P2 (usec)
0	0	0	1	0	0	1	1	Average transmission delay - P3 (usec)
0	0	0	1	0	1	0	0	Maximum transmission delay - P3 (usec)
0	0	0	1	0	1	0	1	Number of Data errors
0	0	0	1	0	1	1	0	Number of RX framing errors
0	0	0	1	0	1	1	1	Number of RX sync errors
0	0	0	1	1	0	0	0	Number of RX WC errors
0	0	0	1	1	0	0	1	Number of bus activity errors
0	0	0	1	1	0	1	0	Number of net initialize errors
0	0	0	1	1	0	1	1	Number of solicit successor cycles
0	0	0	1	1	1	0	0	Number of claim token cycles
0	0	0	1	1	1	0	1	Number of lost tokens detected
0	0	0	1	1	1	1	0	Number of token errors detected
0	0	0	1	1	1	1	1	Number of transmitter errors
0	0	1	0	0	0	0	0	Number of receiver errors
0	0	1	0	0	0	0	1	Number of token hold errors
0	0	1	0	0	0	1	0	Realtime clock drift rate
0	0	1	0	0	0	1	1	Number of abort messages detected
0	0	1	0	0	1	0	0	
		through						
1	1	1	1	1	1	1	1	Reserved

Should the message contain an odd number of records, additional bits of logic 0 data shall be appended to the field sufficient to fill it to a length evenly divisible by 16. The WC field of a REPORT_TERMINAL_STATISTICS message shall define the count of 16-bit word which exist within the field rather than the count of 8-bit records.

3.2.1.1.3.3.8 STATISTICS REPORT

The data field of a STATISTICS REPORT message shall consist of a sequence of 24-bit records, one for each statistic being reported. Bits 0 through 7 shall define the statistic as defined in Table XI. With the exception of the terminal status summary report, bits 8 through 23 shall contain the value of the statistic. Fields reporting numeric statistics shall be organized as a 16 bit binary quantity with the LSB as bit 23.

The Terminal Status Summary report shall be formatted as shown:

Bit 8 : Bit Summary (0 - failed/1 - passed)
Bit 9,10 : Rx Machine Status (1X - Port "A" OK :X1 - Port "B" OK)
Bit 11,12: Tx Machine Status (1X - Port "A" OK :X1 - Port "B" OK)
Bit 13: APM Status (0 - failed/1 - passed)
Bit 14: UIM Status (0 - failed/1 - passed)
Bit 15: Power Status (0 - failed/1 - passed)
Bit 16: Topology Memory (0 - initial condx/1 - operational)
Bit 17-23: Reserved

Should the message contain an odd number of records, additional bits of logic 0 data shall be appended to the field sufficient to fill it to a length evenly divisible by 16. The WC field of a STATISTICS REPORT message shall define the count of 16-bit words which exist within the data field rather than the count of 24-bit records.

3.2.1.1.3.3.9 SET FILTER Message

The data field of a SET FILTER message shall consist of a sequence of 32-bit records, one for each filter word being set. The first word shall consist of the address of the filter word to be set, while the second word shall be the value to which the filter word is to be set. The WC field of a SET FILTER message shall define the count of 16-bit words which exist within the data field rather than the count of 32-bit records.

3.2.1.1.3.3.10 REPORT MESSAGE FILTER

The data field of a REPORT MESSAGE FILTER message shall consist of a sequence of 16-bit records, one for the address of each filter word for which a report is requested.

3.2.1.1.3.3.11 REPORT FILTER Message

The data field of a REPORT FILTER message shall be identical to that described for a SET FILTER message. Only implemented filter words shall be reported.

3.2.1.1.3.3.12 SET REALTIME CLOCK

The data field of a SET REALTIME CLOCK message shall consist of a 48-bit record containing the current time value to be loaded into the realtime clock. The WC field of a SET REALTIME CLOCK message shall define the count of 16-bit words which exist within the record rather than the record count.

3.2.1.1.3.3.13 REALTIME CLOCK REPORT

The data field of a REALTIME CLOCK REPORT message shall be identical to that of a SET REALTIME CLOCK message. The data field shall represent the time value of the local realtime clock at the time the token was received.

3.2.1.1.3.4 Alternate Control Message

The alternate control message capability shall be provided at the discretion of the system designer of a specific HSDB application. It shall be used to specify terminal operations which are judged to be necessary for proper network operation in that application but which are not defined by the HSDB systems specification. It should be noted that alternate control functions will not be compatible across systems boundaries.

3.2.1.1.4 Source Address

The SA field shall consist of a status bit followed by the 7-bit address of the terminal initiating the message. Bit #1 of the field shall be the most significant bit of the address. Bit #0 shall be set to logic "0" except in the case of a ring master terminal. The ring master terminal shall set bit #0 to logic "1".

3.2.1.1.5 Destination Address

The DA shall define the address of the destination terminal(s) for which the message is intended. The first two bits of the FC field define the addressing mode, token (00), maintenance (01), or data (10).

3.2.1.1.5.1 Token Message

Except for CLAIM TOKEN messages, the DA field for a token message shall consist of a "0" in bit #0 followed by the 7-bit address of the terminal for which the token message is intended. Bit #1 shall be the most significant bit of the address.

For CLAIM TOKEN messages, the DA shall consist of the complement of the SA field.

3.2.1.1.5.2 Maintenance Message

The DA field for a maintenance message shall normally consist of a "0" in Bit #0 followed by the 7-bit address of the terminal for which the maintenance message is intended. Bit #1 shall be the most significant bit of the address. Broadcast maintenance messages shall be indicated by DA = "1" in all bits.

3.2.1.1.5.3 Data Message

The DA field for a data message shall define the destination for the message. Bit #"0" of the DA shall define the addressing mode, either singlecast (0) or content/broadcast (1) address, followed by 15 bits as described below.

3.2.1.1.5.3.1 Singlecast Address

The DA field for singlecast mode shall consist of bit #0 set to 0 followed by a 7-bit destination field containing the address of the terminal for which the data message is intended, followed by an 8-bit

subaddress field. The subaddress shall be all zero unless a specific subaddress is explicitly requested by the user at the time the TMU was initiated.

3.2.1.1.5.3.2 Content Address

The DA field for content address mode shall consist of bit #0 set to 1, followed by a 15-bit content descriptor. Each terminal shall maintain a content descriptor filter which shall determine whether or not it wishes to accept a particular content described message. Address FFFF (hexadecimal) shall be the broadcast address.

3.2.1.1.5.3.3 Broadcast Address

Destination Address FFFF (hexadecimal) shall be the broadcast mode address.

3.2.1.1.6 Word Count

The word count field shall describe the number of 16-bit data words which are included in the data field. Word count shall equal the value N-1 where N is the total number of data words (from 1 to 4096 inclusive) included in all records of the message.

3.2.1.1.7 Data

The data field shall consist of one or more records each followed by FCS. A message consisting of more than 256 words shall be decomposed into 2 or more records containing from 1 to 256 (inclusive) words of data. There shall be no restrictions on the format of the data. The terminal shall in no way modify the contents of the data as part of the transmission/reception process except to intersperse FCS as described above.

3.2.1.1.8 Frame Check Sequence

The FCS field shall consist of a 16-bit Cyclic Redundancy Check (CRC) pattern covering the FC + SA + DA + WC + Data fields of the message. CRC shall be calculated based on the following generator polynomial:

$$G(X) = X^{16} + X^{12} + X^{5} + 1 \quad \text{(CCITT-CRC-16)}$$

If the data message from the user contains more than 256 words, the message shall be split into a sequence of 256 word (or remainder) records and an FCS generated for and appended to each record.

3.2.1.1.9 End Delimiter

The ED field shall consist of a 4 bit time sequence of valid and invalid symbols which is the complement of the SD field.

3.2.1.1.10 Maximum Transmission Time

The terminal shall contain provisions to automatically terminate attempted transmissions of greater than the equivalent of a 4096 word data message plus a token message.

3.2.1.1.11 Intermessage Gap

The idle period between adjacent packets on the bus shall be greater than 100 ns and less than 1250 ns.

3.2.1.1.11.1 Terminal Response Time

The terminal shall begin transmission of preamble between 100 and 200ns from detection of a valid message if the message is addressed to the terminal and contains the token.

3.2.1.1.11.2 Receiver Recovery

A receiver shall be capable of synchronizing to and recovering adjacent messages from the bus under minimum intermessage gap times and maximum message dynamic range conditions. A receiver shall be capable of recovering concatenated messages from a single source separated by only ED + SD.

3.2.1.1.12 Redundancy

Two types of network designs shall be supported by the HSDB, single or dual redundant. Networks designed to operate in dual redundant mode shall be implemented using totally independent circuitry for the PHY layer and MED layers. Terminal MAC layer circuitry may be shared if reliability requirements can be met. In either case, a single token (and copy) shall be supported by the network on both buses. The standby bus shall not be used as a separate network. Dual redundant terminals shall be capable of being hard strapped or soft programmed to operate using the "A" bus only.

3.2.1.2 MED Layer Performance

The functions of the MED layer is to provide a data transport highway between terminals. The MED layer, in either coaxial or FO implementation, shall meet the following requirements.

a. Maximum number of terminals - 64
b. Minimum number of terminals - 2
c. Maximum terminal separation - 300 ft
d. Minimum terminal separation - 1 ft
e. Radiated energy - no requirement
f. Susceptibility - no requirement

3.2.1.2.1 Coaxial Media Unique Characteristics

The MED layer, when implemented using coaxial media, shall meet the following requirements.

a.	Stub length - max	- 20 ft
b.	Stub length - min	- 3 ft
c.	Stub attenuation - max	- 2.0 db max at 50 MHz
d.	Stub characteristic impedance	- 50 ohms ±2.0 ohms
e.	Mainbus impedance	- 50 ohms ±1.25 ohms
f.	Mainbus attenuation - max	- 13 db total at 50 MHz
g.	Coupler loss - mainbus in/out	- 0.100 db max
h.	Coupler loss - mainbus to Rx	- 25.75 db ±0.5 db
i.	Coupler loss - Tx to mainbus	- 10.75 db ±0.5 db
j.	Coupler Tx port impedance	- 50 ohms ±5 ohms
k.	Coupler Rx port impedance	- 50 ohms ±5 ohms
l.	Connector loss(per mated pair)	- 0.011 db
m.	Propagation delay - mainbus	- LT 500 ns
n.	End to end loss - max	- 54 db
p.	Coupler mainbus return loss	- NLT 32 db at 50 MHz
q.	Differential "A"/"B" skew	- NMT 400 ns

A simplified system signal budget diagram is shown in figure 2a. Note that the end to end loss specification is consistent with a 64 terminal network of 300 ft overall length constructed using mainbus cable exhibiting a loss of 1.8 db/100 ft at 50 MHZ. Networks of longer length shall be possible given fewer terminals or lower loss mainbus cable, within the mainbus loss specification. Networks of more than 64 terminals shall be possible given shorter mainbus and stub lengths and/or lower loss cable, within the system loss specification.

3.2.1.2.2 FO Media Unique Characteristics

The MED layer, when implemented using FO media, shall meet the following requirements.

a.	Stub attenuation - max	- 4.5 db/km @ 850 nm
b.	Coupler attenuation - max	- 21.5 db any input to any output
c.	Stub cable propagation	- Graded index
d.	Stub cable bandwidth	- GT 30 MHz/km
e.	Stub diameter (core/cladding)	- 100/140 microns
f.	Stub numerical aperture	- 0.29 ±0.015
g.	Stub propagation constant	- GT 0.63
h.	Differential "A"/"B" skew	- NMT 400 ns

A simplified power budget diagram is shown in figure 2b. The overall loss specification is consistent with a 64 port star coupler centered in a network topology of 300 ft overall length with a single connector per stub. Networks of longer overall length are possible if a coupler with fewer than 64 ports is used, within the constraints of the overall system loss budget.

3.2.1.3 PHY Layer Performance

The function of the PHY layer is to translate data presented in message form from the MAC layer into a modulated signal (packet) compatible with the transport characteristics of the MED layer and to demodulate packets from the MED layer back into message form for presentation to the MAC layer. The PHY layer shall be implemented in

the form of a Transmit/Receive Unit (TRU) contained within each terminal. Figure 6 illustrates the functional characteristics of the TRU. Requirements common to both types of TRU (coaxial and FO) are stated below.

a.	TX Bit rate	- 50 Mbps ±0.0125 Mbps
b.	RX acquisition range	- 50 Mbps ±0.025 Mbps
c.	Preamble (clock sync)	- 8 bits
d.	Start delimiter	- 4 bit times
e.	End delimiter	- 4 bit times
f.	Transmitter timeout	- 1.3 ms + 0.13 ms/-0.0 ms
g.	System error rate	- LT 5X10E-11 BER Over operating envelope

Note that for dual redundant terminals, a single transmit message from the MAC layer results in two packets being sent, one on the "A" port and one on the "B" port. Conversely, two packets, one from RX-"A" port and another from the RX-"B" port result in two received messages being sent to the MAC layer.

3.2.1.3.1 Coaxial TRU

The coaxial TRU shall provide the electrical and physical interface of each terminal with the MED layer of a coaxial implementation of the HSDB. Requirements unique to the coaxial TRU are stated below.

a.	Modulation	- Manchester II
b.	Tx output level	- 15.0 ±1.0Vp-p into 50ohms
c.	Tx polarity	- negative transition at mid bit = logic "1"
d.	Bit symmetry	- 50% ±10%
e.	Tx switch output level - Tx state	- 11 ±4 into 50 ohms
f.	Tx switch output level - Rx state	- -2.5 ±0.25 Vdc into 50 ohms
g.	Tx switch risetime/falltime	- LT 100 ns
h.	Rx input sensitivity	- 25 mV p-p for stated error rate
j.	Rx dynamic range	- NLT 27db
k.	Input impedance	- 50 ohms ±5 ohms
l.	Clock sync	- To within 3 ns within 6 bits
m.	Rx polarity	- Negative transition at mid bit = logic "1"

3.2.1.3.2 FO TRU

The FO TRU shall provide the electrical and physical interface of each terminal with the MED layer of a FO implementation of the HSDB. Requirements unique to the FO TRU are stated below.

a.	Modulation	- Manchester equivalent
b.	Tx output level	- 200uw-Pk to 400uw-Pk

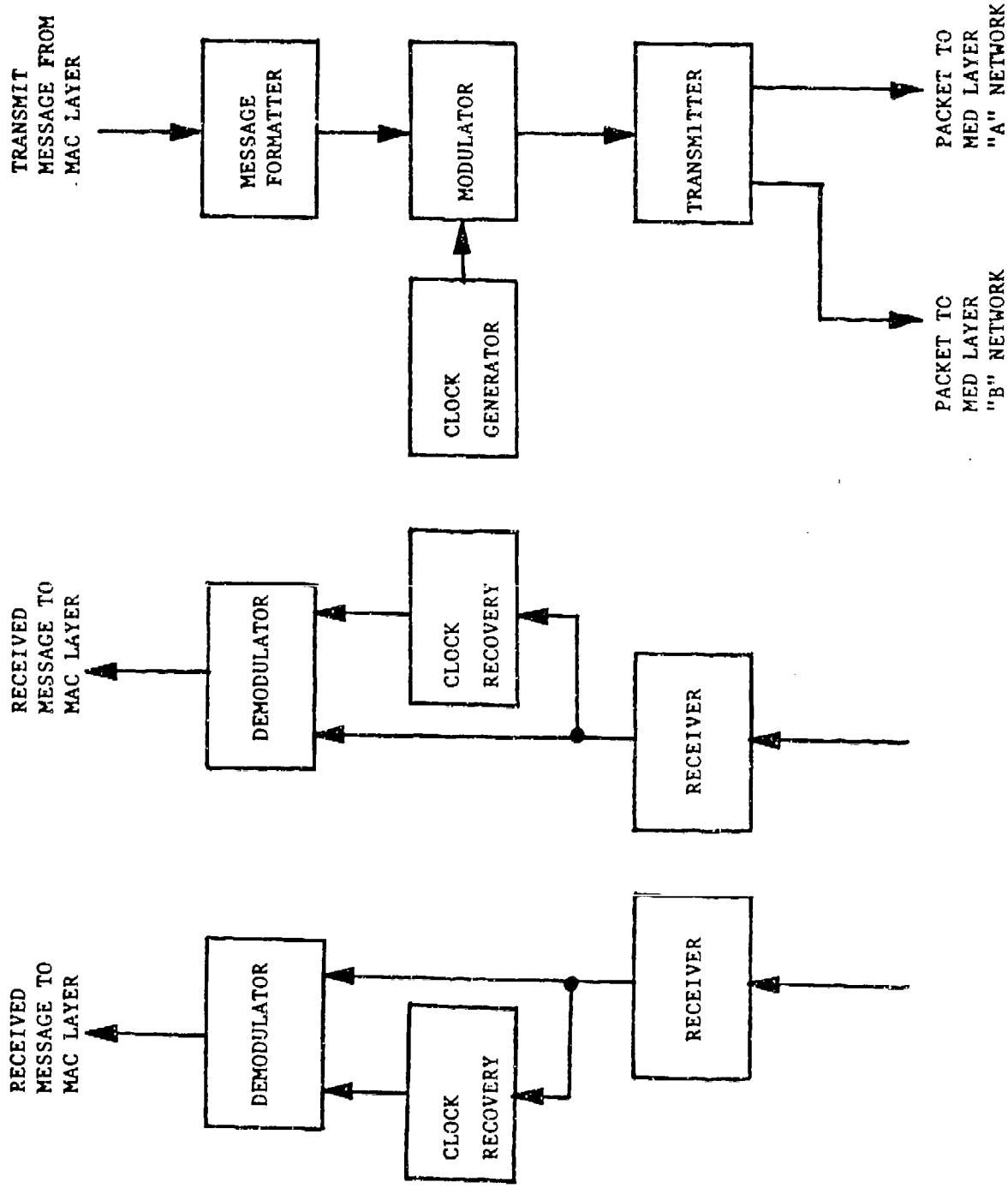


FIGURE 6
PHY LAYER SIMPLIFIED FUNCTIONAL DIAGRAM

c.	Tx wavelength	- 850 nm ±30 nm median
d.	Tx spectrum	- GT 75% of the power shall be within 30nm of median wavelength
e.	Tx polarity	- Power in first 1/2 bit period = logic "1"
f.	Tx bit symmetry	- 50% ±10%
g.	Tx rise/fall time	- LT 4ns
h.	Rx sensitivity	- GT 1uw-Pk for stated error rate
j.	Rx dynamic range	- 21db or greater
k.	Clock sync	- To within 3ns within 6 bit periods
l.	Rx polarity	- Power in first 1/2 bit period = logic "1"
m.	Rx Rise/full time	- LT 5ns for stated error rate

3.2.1.4 MAC Layer Performance

The function of the MAC layer is to manage participation of the terminal in network transactions including data transmission and reception, token management, and network maintenance functions. The MAC layer shall perform all protocol functions of the terminal in a manner so as to appear transparent to higher layers of control within the terminal.

A token passing protocol shall be implemented by the MAC. The token address of each terminal shall be identical to the terminal address (TA) hardwired into each terminal. Each terminal shall have an unique TA.

a. A token shall control the right of access to the physical medium; the terminal which holds the token shall have singular momentary control over the medium.

b. The token shall be passed among terminals residing on the network in ascending order of their TAs, thus sharing network bandwidth in a planned manner. A logical ring control structure shall thus be superimposed on the linear physical structure of the network.

c. Nominal operation of the network shall consist of a data transfer phase (optional) and a token transfer phase on the part of each terminal during each network rotation.

d. Token maintenance functions shall reside in each and every terminal for the purposes of initializing the logical ring, lost token recovery, station deletion, new station addition, and general housekeeping.

The MAC shall perform as a loosely coupled set of four logical machines as shown by figure 7. Data and control/status information shall be communicated into and out of the MAC in the form

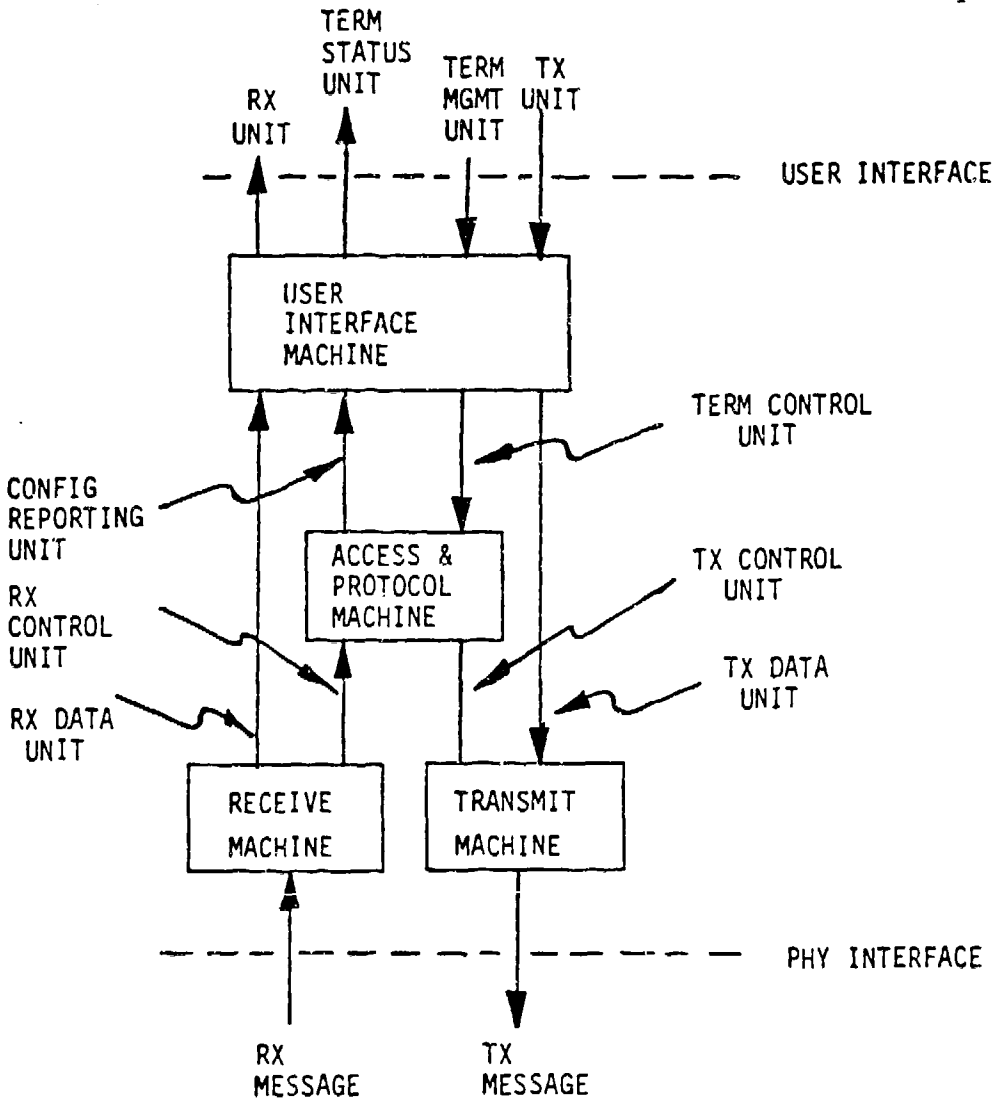


FIGURE 7

MAC LAYER FUNCTIONAL DESCRIPTION

of a set of transaction data units. TX Units and Terminal Management Units shall be operated upon to create TX messages for presentation to the PHY layer. RX messages from the PHY layer shall be decoded so that only properly received RX Units are presented to the User. Upon request, the MAC shall present Terminal Status Units to the User.

3.2.1.4.1 **Transaction Data Unit Descriptions**

Data and control/status information shall be communicated into and out of the MAC in the form of a prescribed set of data units. Communications among the functional entities within the MAC shall also be accomplished in this manner. Figure 7 illustrates the twelve different units described in subsequent paragraphs and shows the source and destination of each.

3.2.1.4.1.1 TX Unit (TXU)

The TXU shall consist of the data block to be sent from the user of the terminal to the user of some other terminal(s). The data shall be organized as a block of from 1 to 4096 16 bit words.

3.2.1.4.1.2 Terminal Management Unit (TMU)

The TMU shall consist of a data set sent from the user to the terminal. It shall be used to set the operating mode of the terminal, to request transmission of a data block, or to request network/terminal maintenance information. Unique identifiers shall be provided to accommodate the functions described in Table XII.

TABLE XII
TMU/TCU FUNCTIONS

Function	Requirements
Token management	6 bits
Request to send	3 bits
Class of service	1 bits
Priority	2 bits
Destination	16 bits
Status request	6 bits
RX message received	6 bits
Word count	12 bits

3.2.1.4.1.3 TX Data Unit (TXD)

The TXD shall consist of the data block presently being sent by the terminal. It shall consist of a sequence of 16 bit words clocked from the MAC at a rate consistent with the operating bus rate of the terminal.

3.2.1.4.1.4 TX Control Unit (TXC)

The TXC shall consist of header information appended to the TXD prior to transmission on the network. It shall consist of the sequence of functions described in Table XIII and shall be clocked

from the MAC at a rate consistent with the operating bus rate of the terminal.

TABLE XIII
TXC FUNCTIONS

Function	Requirement
Frame control	8 bits
Source address	7 bits
Destination address	16 bits

When the terminal is operating in maintenance re-transmit mode, TXC shall consist of the turnaround message to be placed on the bus.

3.2.1.4.1.5 Terminal Control Unit (TCU)

The TCU shall consist of a set of data bits which define the operating mode of the terminal, to request transmission of a data block, or to request network/terminal maintenance information. The functions described in Table XII shall be accommodated.

3.2.1.4.1.6 TX Message (TXM)

The TXM shall consist of a serial data stream comprising the message to be sent by the terminal, in the format "FC + DA + SA + WC + Data + FCS." The data stream shall be clocked from the MAC layer at the rate and time required to support the operating bus rate of the terminal.

3.2.1.4.1.7 RX Message (RXM)

The RXM shall consist of a serial data stream comprising the message being recovered by the terminal. It shall be in the same format as present on the network without the ED and SD fields but including the "FC + DA + SA + Data + FCS" fields.

3.2.1.4.1.8 RX Data Unit (RXD)

The RXD shall consist of the contents of the "Data" field of the message just received from the network. RXD shall be initiated only after FCS has been validated against the frame contents and the destination address is found to correspond with a valid destination address.

3.2.1.4.1.9 RX Control Unit (RXC)

The RXC shall consist of a bi-directional data path used to coordinate the reception of messages. Valid destination address(es) shall be passed from the access and protocol machine to the receive machine (refer to figure 5). The "FC + SA" fields from each received message shall be passed from the receive machine to the access and protocol machine.

3.2.1.4.1.10 RX Unit (RXU)

The RXU shall consist of a data block comprising the "Data" field of a single message. It shall be passed from the MAC layer to the USER at the convenience of the user. The data shall be organized as a block of from 1 to 4096 16 bit binary words.

3.2.1.4.1.11 Configuration Reporting Unit (CRU)

The CRU shall be used to report information relative to network and terminal operations. Unique identifiers shall be provided to accommodate the functions described in Table XIV plus a minimum of 50% spares.

TABLE XIV
CRU FUNCTIONS

Function	Requirements
Terminal configuration	16 bits
Terminal status	16 bits
Network configuration	Per 3.2.1.4.2.12
Statistics	Per Table XI

3.2.1.4.1.12 Terminal Status Unit (TSU)

The TSU shall be initiated upon receipt of a valid message from the bus or upon request of the user. In the former case, it shall inform the user of the message location, size, source, class of service, and priority. In the latter case, it shall respond to the specific request of the user.

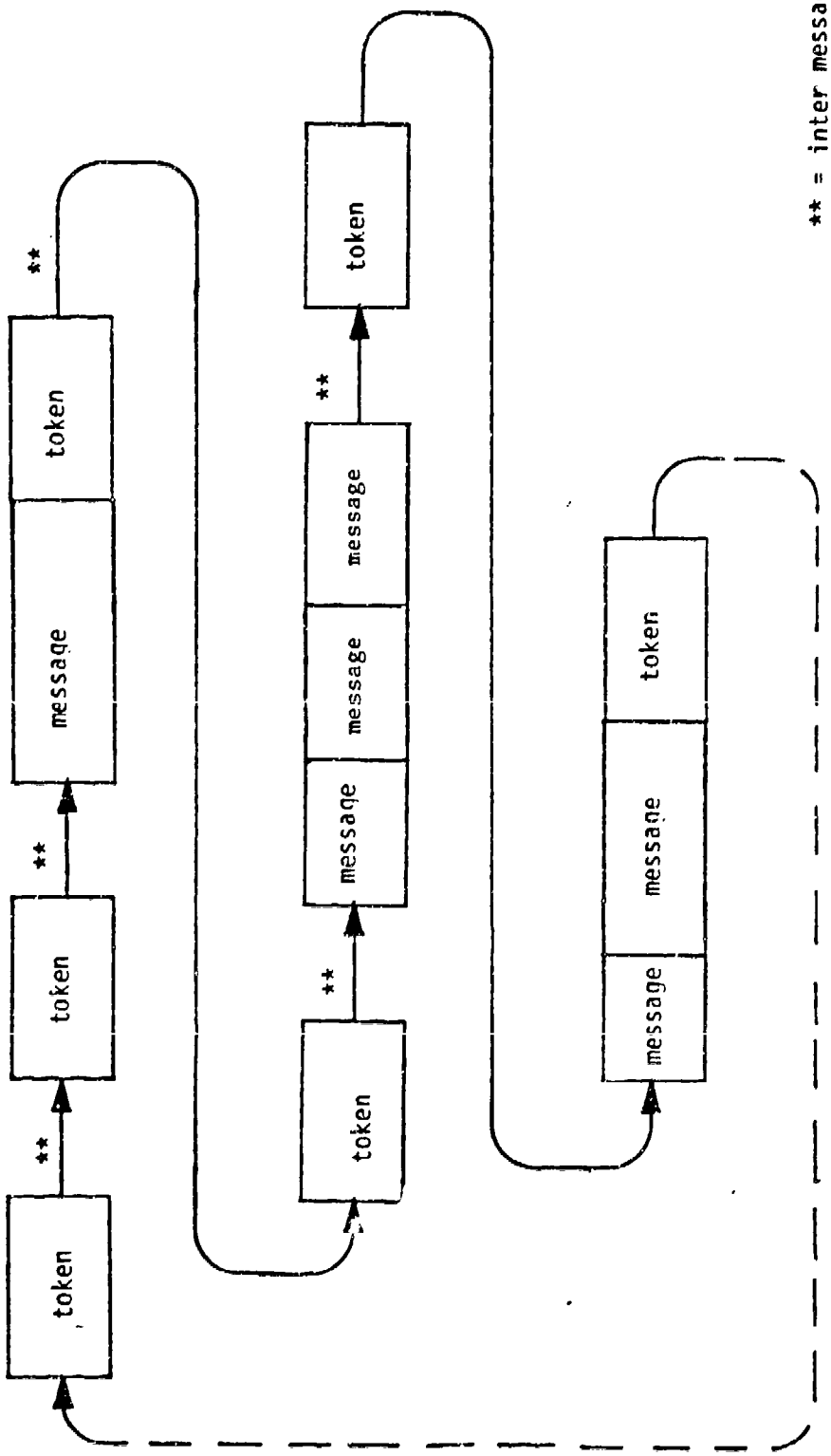
3.2.1.4.2 Functional Requirements

The MAC layer shall perform as a loosely coupled set of four logical machines as described in figure 7. The heart of the MAC is the access and protocol machine. Its function is the establishment and maintenance of the network and coordination of other functions within the MAC.

3.2.1.4.2.1 Steady State Network Operation

Steady state network operation shall require the sending of the token to a specific successor when each terminal is finished transmitting scheduled traffic, if any. The sequence of messages involved in information transfer is illustrated in Figure 8. After sending the token, the terminal shall listen for evidence that its successor has passed the token. If the token sender does not hear evidence that the token pass was successful, it shall begin a series of recovery procedures that grow increasingly more drastic as described in 3.2.1.5.2. If all attempts prove unsuccessful, the terminal shall become silent and wait for another terminal to initiate recovery procedures.

3.2.1.4.2.2 Deleted



** = inter message gap

Figure 8
TOKEN PASSING MODE INFORMATION TRANSFER FORMAT

3.2.1.4.2.3 Realtime Topology Adaptation

On a periodic basis, and when the level of network traffic is sufficiently low, new terminals shall be allowed access to the network. The sequence of transactions is described in Table XV. The process shall be initiated by the terminal in possession of the token. It shall take the form of a poll to one potential TA between its own TA and the TA of its successor. If the poll is successful, the logical ring shall be broken in the proper TA sequence and the requesting terminal shall be inserted.

TABLE XV
TOPOLOGY ADAPTATION TRANSACTION SEQUENCE

Soliciting Terminal	New Terminal	Previous Successor
Solicit Entry ---------->*		
*<---------- Request Entry		
Set Successor ---------->*		
Set Predecessor ---------->*		
Pass Token ---------->*		

3.2.1.4.2.4 Initialization Process

Initialization of the network shall be a special case of adding new terminals to an existing network. Each potential initializer shall send a frame having a length proportional to its address. The terminal shall then listen to the bus. If all other potential initializers had finished transmitting prior to that time, the terminal will hear a quiet bus and assume that it holds the only token in the network. It will then proceed to form the network by a polling technique.

3.2.1.4.2.5 Priority

A 4-level priority mechanism shall be provided for use at the discretion of the network designer. The mechanism shall reserve a set amount of network bandwidth for messages at each priority level. This shall be accomplished by using a set of three token rotation timers in each terminal. The terminal shall send only messages of a priority higher than indicated by the timers, based on the time interval since it last held the token.

3.2.1.4.2.6 Maintenance Functions

Maintenance functions shall be provided to allow the quality of service being experienced by any terminal to be reviewed by its user or by any other user in the network. A multilevel maintenance strategy shall be provided in each terminal as described below:

a. Built-In-Test (BIT) shall constantly monitor critical terminal functions.

b. The terminal receiver shall monitor all terminal transmissions.

c. A local loopback mode shall be provided wherein the user shall be able to send a test message between transmitter and receiver without the message appearing on the network.

d. A network loopback mode shall be provided wherein any terminal shall be able to originate a test message to itself from another terminal on the network. Interaction of the users of either terminal shall not be required.

Performance statistics shall be computed within each terminal describing the message delays for each priority level, the number of erroneous messages received, average and peak values of the token cycle times and other parameters of significance to the system designer. Registers associated with statistics shall be reset each time the statistic is reported. Registers which reach overflow state before being reported shall retain the maximum count until reported.

3.2.1.4.2.7 Realtime Clock

A 48-bit realtime clock (RTC) shall be established within each terminal at initialization. The clock may be set by the user, set from the network, read by the user, or read from the network. The RTC shall be accurate to ±120 micro second over a period of 1 second without drift correction. Clocks within different terminals on the network shall be synchronized to within 10 microseconds using drift rate correction and 1 second update rate. Clocks within different terminals on the network shall be synchronized to within 3 microseconds using drift rate correction, differential delay correction and 1 second update rate. Provisions shall be included to allow the embedded clock of any terminal to be defined as the master clock. It shall also be possible to synchronize the master clock to some high accuracy time reference located external to the terminal.

3.2.1.4.2.8 Acknowledgment

Automatic acknowledgment of messages is not a function of the HSDB. Whenever acknowledgment is required to meet system performance needs, the function shall be provided by the user.

3.2.1.4.2.9 Retry

Automatic retry of token pass messages shall be in accordance with the applicable paragraphs of this specification. Automatic retry of data and maintenance messages is not a function of the HSDB. Whenever required to meet system performance needs, the function shall be provided by the user.

3.2.1.4.2.10 Redundant Bus Operation

A dual path redundant network design shall be the preferred implementation approach. A single path network shall be non-preferred but shall be accommodated within the design of dual bus terminals. A single bus terminal shall be a non-preferred case but shall be accommodated within a dual path redundant network.

Redundant terminals shall transmit and receive on both the "A" bus and the "B" bus at all times thus effectively providing parallel equivalent transmission paths throughout the network. The message first arriving at the terminal shall be recovered unless it contains one or more transmission errors. In that case, the later arriving message shall be recovered.

3.2.1.4.2.11 Message Filter

Terminals shall accept content addressed messages. A content descriptor message filter shall be used for this purpose. The content descriptor filter shall consist of a series of $(2^{15}-1)$ bits which are either set or clear, depending on whether (1) or not (0) the terminal is to receive the data contained within the message. The filter is organized as an array of up to 2^{11} words of 16 bits each. Content address 0 corresponds to word #0, bit #0. Content address 1 corresponds to word #0, bit #1. In general, content descriptor N shall correspond to the word described by the integer portion of (N/16) and the bit described by (N MOD 16).

3.2.1.4.2.12 Topology Memory

Each terminal shall include provisions for a topology memory function for use in error detection and recovery. The topology memory shall accommodate 128 terminal records. All but the "this terminal" record shall contain fields as described below.

a. Terminal Address (TA): Range - 0 through 127
b. Current Predecessor Address (CP): Range - 0 through 255; Initial Value - 255 (error)
c. Current Successor Address (CS): Range - 0 through 255; Initial Value - 255 (error)
d. Status: Range - on-line (11), off-line (01), absent (00); Initial Value - ABSENT
e. Operational Ports: XXX1 - Receive on "A"; XX1X - Receive on "B"; X1XX - Transmit on "A"; 1XXX - Transmit on "B"; Initial Value - 0000
f. Ringmaster: Range - NOT RINGMASTER (0), RINGMASTER (1). Initial Value - 0
g. Timekeeper: Range - NOT TIMEKEEPER (0), TIMEKEEPER (1). Initial Value - 0

Except as noted below, the topology memory shall be set to its initial state each time power is applied. The status field shall be settable either by the user or from the network via a SET_TOPOLOGY

message. The predecessor address and successor address fields shall be settable from the network via a SET_PREDECESSOR and SET_SUCCESSOR message, respectively, or by a SET_TOPOLOGY message.

The "this terminal" entry in the topology memory shall be identical to those listed above and shall be controlled in the same manner except:

a. The terminal address shall be hardwire strapped in the terminal.

b. Status shall be setteble from either the user function or from the terminal itself. The initialization value shall be "OFF-LINE." The value "ABSENT" is undefined.

c. Enable bus shall be set based on the type of terminal (single bus/dual bus) and the results of BIT.

3.2.1.4.2.13 Network Activity Monitor

Each terminal shall include a network activity monitor which shall trigger whenever a signal is being received from the network. The monitor shall derive its determination of network activity from measurement of RMS power at the receive stub port of the terminal as defined below.

a.	Sensitivity - wire bus terminal	5uw
b.	Sensitivity - FO terminal	0.5uw
c.	Dynamic range - wire bus terminal	30db
d.	Dynamic range - FO terminal	24db
e.	Response Time	60ns
f.	Recovery time (from maximum signal)	140ns

Dual redundant terminals shall include an independent monitor function for each RX port.

3.2.1.4.3 Timers

Each terminal shall include 10 timer functions used to coordinate the various functional operations of the terminal. Their relationship to other terminal functions is illustrated by figure 9. The clock of each timer except the Retry Limit Timer shall be derived from the terminal bit clock (nominal 50 MHz). Each timer shall be provided with a default initialization value as defined below. The default value may be overridden by a value received from that terminal's user or by a value received from remote user via of a SET PARAMETER message.

3.2.1.4.3.1 Token Hold Timer

A token hold timer (THT) is provided to limit the period of time over which a terminal may schedule traffic. The THT shall be reset to its initialization value whenever the terminal receives a token. It shall count toward zero at a rate of 0.32uS (bit clock*16) so long as the terminal holds the token. The default initialization

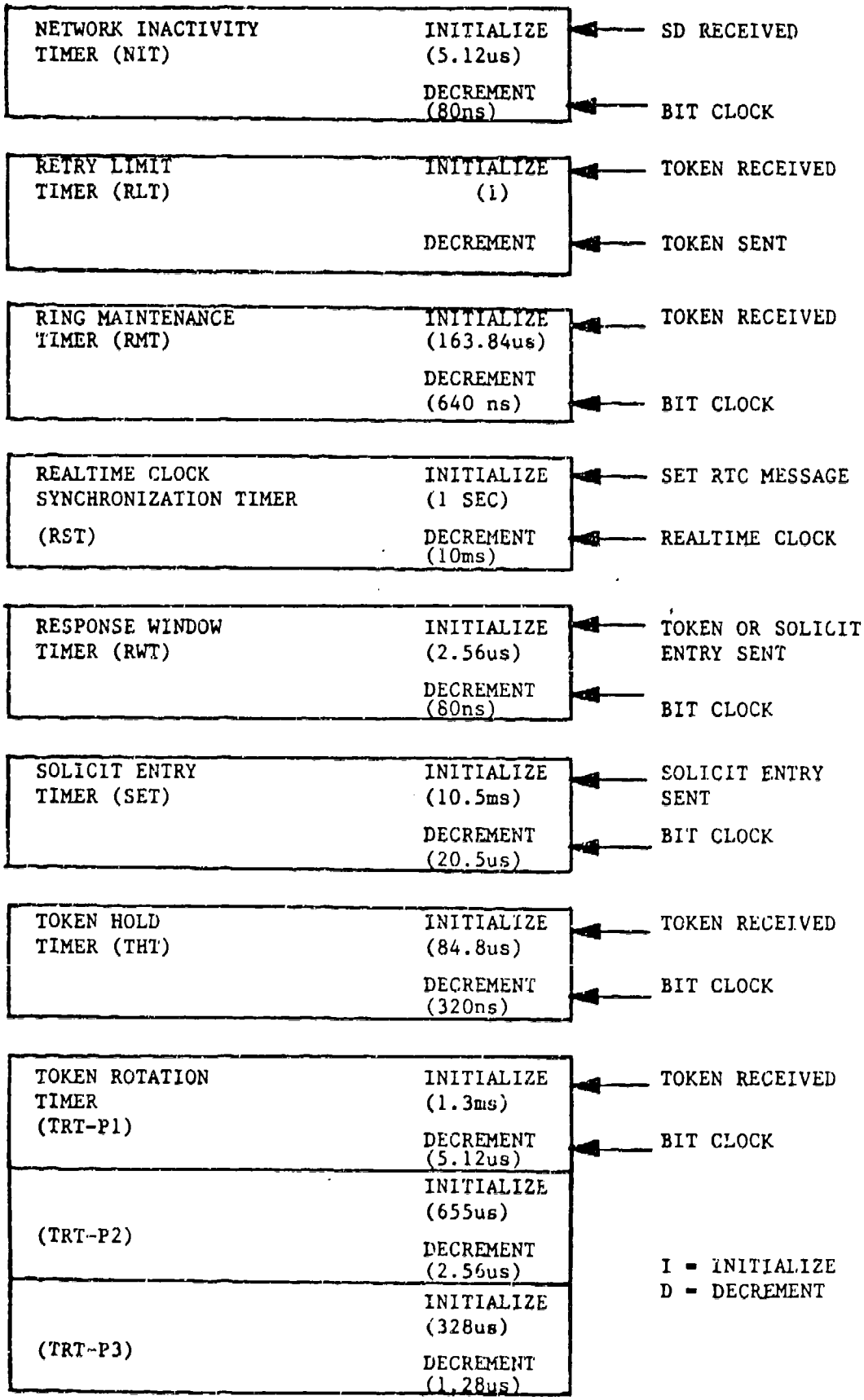


FIGURE 9
TERMINAL TIMER RELATIONSHIPS

value shall be 265 word times (84.8us). Programmable initialization values from 5 word times (1.6us) through 4119 word times (1318us) shall be accepted.

3.2.1.4.3.2 Token Rotation Timers

Three 14-bit token rotation timers (TRT) are provided to control the terminal's access to the network at each priority level. Each TRT shall be set to its initialization value each time the token is received. It shall begin to count toward zero immediately thereafter.

a. TRT-P1 shall be used to govern transmission of P1 priority messages (high priority). The clock rate for TRT-P1 shall be 5.12us (bit clock*256). The default initialization value shall be 1.31072ms (rate clock *256). The initialization value shall be programmable over the range 81.92us (rate clock * 16) through 83.881ms (rate clock * 2^{14}).

b. TRT-P2 shall be used to govern transmission of P2 priority messages. The clock rate for TRT-P2 shall be twice that of TRT-P1. The default initialization value shall be 655.36us (rate clock * 256). The initialization value shall be programmable over the range 40.96us (rate clock * 16) through 41.940ms (rate clock * 2^{14}-1).

c. TRT-P3 shall be used to govern transmission of P3 priority messages (lowest priority). The clock rate for TRT-P3 shall be twice that of TRT-P2. The default initialization value shall be 327.68us (rate clock * 256). The initialization value shall be programmable over the range 20.48us (rate clock * 16) through 20.970ms (rate clock * 2^{14}-1).

3.2.1.4.3.3 Ring Maintenance Timer

Each terminal shall include a 14-bit ring maintenance timer (RMT) function which is used to defer realtime topology adaptation while the bus is heavily loaded. The RMT shall be set to its initialization value each time the token is received. It shall count toward zero until the next reset occurs. The clock rate shall be twice that of TRT-P3. The default initialization value shall be 163.84us (rate clock * 256). The initialization value shall be programmable over the range 10.24us (rate clock * 16) through 10.486ms (rate clock * 12^{14}-1).

3.2.1.4.3.4 Solicit Entry Timer

Each terminal shall include a 16-bit solicit entry timer (SET) function which is used to indicate to the terminal that a new successor shall be sought. SET shall be set to its initialization value each time a successor is solicited, whether said solicitation is successful or not. The timer shall be decremented at a rate of 20.48us (bit clock * 1024) until the next reset occurs or until reaching zero

count. The default initialization count shall be 10.5ms (rate clock * 512). The initialization value shall be programmable over the range 20.48ms (rate clock * 2^0) through 1.34 sec (rate clock * 2^{16}-1).

3.2.1.4.3.5 Network Inactivity Timer

Each terminal shall include an 8-bit network inactivity timer (NIT) which is used to indicate that the network has been idle for an unexpected period. The NIT shall be set to its initialization value each time the network activity monitor indicates that the network is idle. It shall count toward zero at a rate of 80ns (bit clock * 4) until the counter reaches zero (network inactivity error) or until the monitor recognizes activity on the network. The default initialization count shall be 5.12us (rate clock * 64). The initialization value shall be programmable over the range 320ns (rate clock * 4) through 20.40us (rate clock * 255). Dual redundant terminals shall implement a single NIT function for both Rx ports. Activity on a single port shall be sufficient to reset the NIT.

3.2.1.4.3.6 Response Window Timer

Each terminal shall include a 7-bit response window timer (RWT) which is used to determine when a token has been lost or when a SOLICIT ENTRY message has had no response. The RWT shall be set to its initialization value each time the terminal terminates (ED) a packet. It shall count toward zero at a rate of 80ns (bit clock/4) until the counter reaches zero (no-response error) or until the terminal recognizes a message with SA = the destination address of its previous message (successful response). The default initialization count shall be 2.56us (rate clock * 32). The initialization value shall be programmable over the range 320ns (rate clock * 4) through 10.16us (rate clock * 127). Dual redundant terminals shall implement a single RWT function. A valid response received from either RX port shall terminate the countdown. The terminal design shall ensure that RWT is always set shorter than NIT.

3.2.1.4.3.7 Retry Limit Timer

Each terminal shall include a 3-bit retry limit timer (RLT) which is used to determine the number of retries on a token pass before instituting the token recovery process. The RLT shall be set to its initialization value each time the token is received. It shall count toward zero at a rate of 1 count per token pass attempt until the counter reaches zero (token pass failure) or until the transaction is successful. The default initialization value shall be 1. The initialization value shall be programmable over the range 1 through 7.

3.2.1.4.3.8 Realtime Clock Synchronization Timer

Each terminal shall include a 10-bit realtime clock synchronization timer (RST) which is used to maintain continuity of the realtime clock update function. The RST shall be set to its initialization value each time a RTC maintenance message is received. It shall count toward zero at a rate of 10ms until the counter reaches zero or until reset. The initialization value shall be programmable

over the range 100ms through 10 sec. Two default initialization values shall be provided, TIMEKEEPER (TK) = 1.0 sec, and not TIMEKEEPER (TKF) = 2.0 sec.

3.2.1.4.4 State Machine Descriptions

The MAC layer shall perform as a loosely coupled set of four state machines. The physical and electrical implementation of a terminal shall be at the discretion of the terminal designer so long as the functional requirements described herein are met, so as to ensure the compatibility of different terminal designs.

3.2.1.4.4.1 Access and Protocol Machine (APM) Description

Normal operation of the APM function shall conform to the architecture described in figure 10. Operation of the terminal under error conditions shall be as described in 3.2.1.5.

3.2.1.4.4.1.1 Off-Line State

The terminal shall enter off-line (state 1) immediately following power up or following detection of certain fault conditions. This state shall be limited to self test functions which do not transmit on the network. After completing self test, the terminal shall remain in state 1 until all initialization parameters have been received from the user and it has been instructed to go on-line. That sequence shall cause the terminal to transition to the IDLE state.

3.2.1.4.4.1.2 IDLE State

While in IDLE (state 2), the terminal shall monitor the network for activity. Dependent upon network traffic intercepted, the appropriate next state shall be entered as follows:.

a. If the NIT expires with no activity monitored on the network, the CLAIM_TOKEN state shall be entered.

b. If the terminal is on-line and a token message addressed to that station is received, the transition shall be to USE_TOKEN state.

c. If a SOLICIT_ENTRY message of the terminal's address is received and the terminal desires entry to the network, the ENTER_NET state shall be entered.

d. If a built-in-test fault is noted, the OFF_LINE state shall be entered.

3.2.1.4.4.1.3 USE_TOKEN State

USE_TOKEN (state 3) is the state from which the terminal shall transmit data messages on the network. It shall enter state 3 from state 2 or state 5 whenever it receives a valid token from the network. It shall enter from state 6 when it has claimed the token from an initialization process.

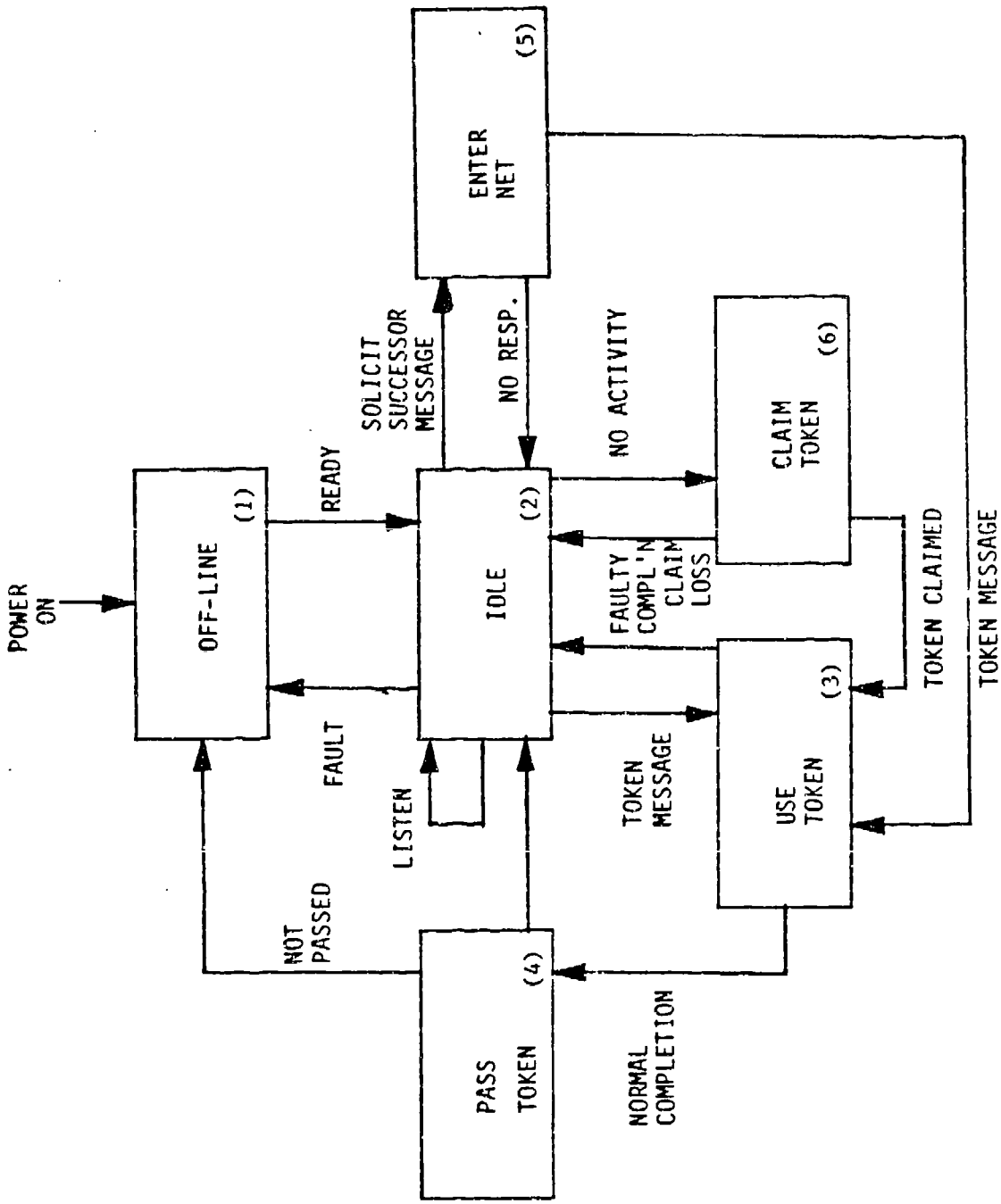


FIGURE 10

APM STATE DIAGRAM

Exhaustive class of service shall be the preferred and default mode of operation for HSDB terminals. The process is illustrated in figure 11.

a. The transmit queue shall be tested. If empty, the scheduled traffic shall be sent and the terminal shall transition to PASS_TOKEN state. Note that the PASS_TOKEN state is outside the jurisdiction of the THT.

b. Assuming that the queue is not empty, examine the top (oldest) message in the queue. If it will fit within the remaining schedule, move it from the queue to the schedule.

c. Calculate the THT remainder.

d. Increment the queue to the next oldest message.

e. Repeat steps a through d until the entire queue has been examined or until the THT has been exhausted.

When using multiple level priority class of service, the USE_TOKEN process shall progress in accordance with figure 11.

a. The three TRTs shall be tested and the appropriate flag (F1, F2, F3) either set (TRT active) or cleared (TRT timed out).

b. The TRTs shall be reset to their initialization value and shall begin decrementing.

c. The first message in the P0 queue shall be placed in the schedule and the length of the message shall be deleted from the THT register.

d. The next message in the P0 queue shall be checked against the THT remainder. It shall be scheduled as described in step c above if its length is less than the value of THT remainder.

e. The process shall continue as described above for the remainder of the P0 message queue.

f. F1 shall be checked. If set, messages at the P1 level shall be scheduled as described for P0 in steps c through e above.

g. Similarly, lower priority messages (P2 and P3) shall be scheduled so long as the flags and THT allow.

h. Scheduled traffic shall be sent in the order scheduled (highest priority first).

i. The terminal shall transition to the PASS_TOKEN state.

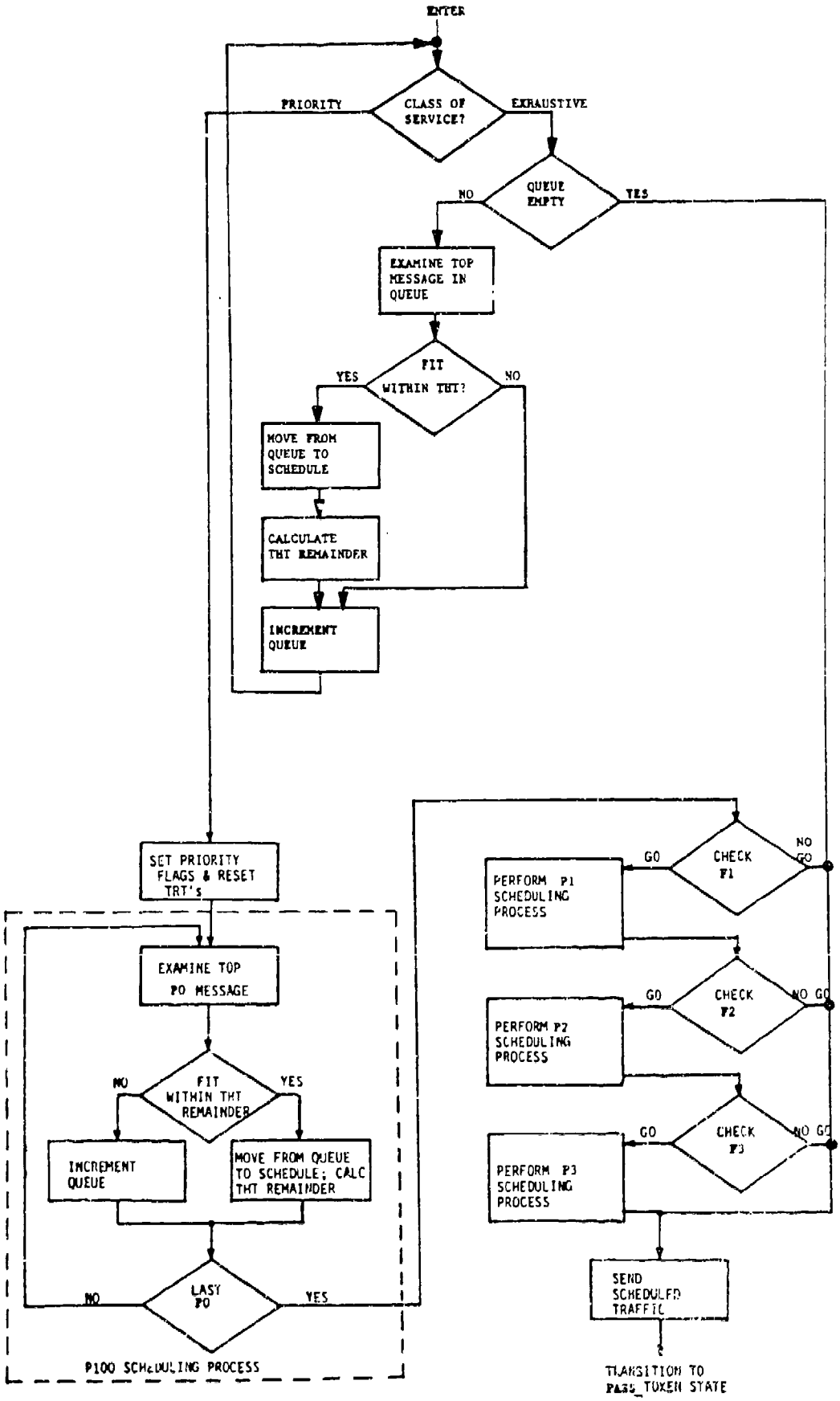


FIGURE 11
USE_TOKEN CONTROL FLOW DIAGRAM

3.2.1.4.4.1.4 PASS_TOKEN State

PASS_TOKEN (state 4) is the state from which control of the network is relinquished to a successor terminal. It shall only be entered from the USE_TOKEN state and shall always transition to the IDLE state.

The PASS_TOKEN process shall progress in accordance with figure 12.

a. The ring maintenance timer (RMT) shall be tested. If it has timed out, the token shall be passed without soliciting for a new successor.

b. The Solicit Entry Timer (SET) shall be checked. If inactive, the process to look for a new successor shall be initiated. If the SET is still active, the token shall be passed without soliciting for new terminals.

c. The network topology map shall be checked to confirm that one or more potential successors exist. If the current successor (CS) equals the terminal address +1, then no potential successors exist and the token shall be passed without soliciting for new terminals.

d. Assuming the three tests (above) have passed, the terminal shall perform the SOLICIT ENTRY process as described in 3.2.1.4.4.1.7.

e. The terminal shall pass the token to its successor.

3.2.1.4.4.1.5 ENTER_NET State

The ENTER_NET (state 5) state shall be initiated upon command of the terminal user and upon receipt of a SOLICIT_ENTRY message from the network. As illustrated by figure 13, the terminal shall respond to the SOLICIT_ENTRY message by sending a REQUEST_ENTRY + TERMINAL_STATUS_SUMMARY maintenance message. Assuming that the soliciting terminal recognizes the REQUEST_ENTRY message, it shall splice the new terminal into the logical ring using SET_SUCCESSOR and SET_PREDECESSOR messages and then pass the token to the NEW terminal which will transition to the USE_TOKEN state. The normal sequence of transactions is described in Table XV.

If the next token pass message on the network is not addressed to the terminal requesting entry then that terminal shall assume that its claim was not recognized and will transition to the IDLE state.

3.2.1.4.4.1.6 CLAIM_TOKEN State

The CLAIM_TOKEN state (state 6) shall be entered from the idle state when the Network Inactivity Timer (NIT) expires. The terminal shall attempt to initialize (or reinitialize) the ring using the procedure described in figure 14.

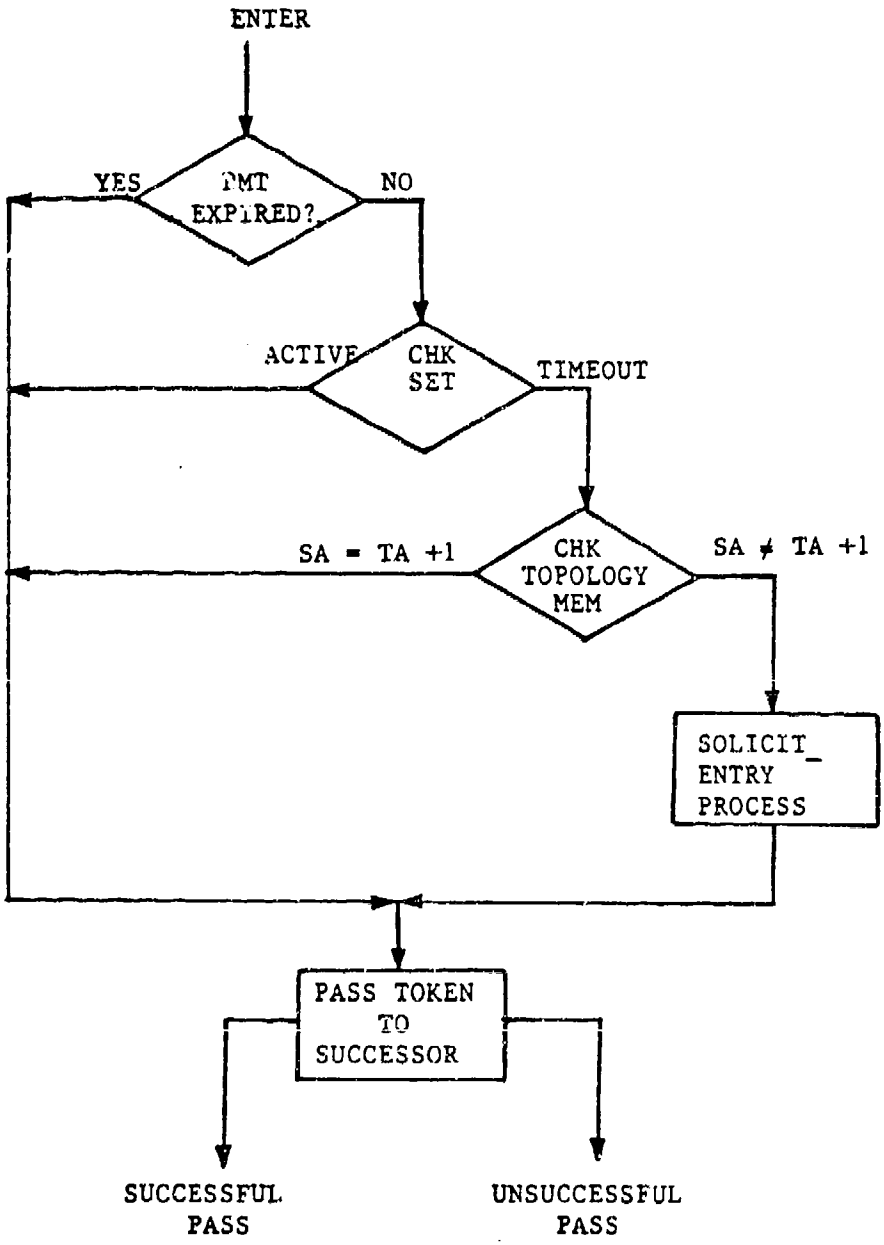


FIGURE 12
PASS_TOKEN CONTROL FLOW DIAGRAM

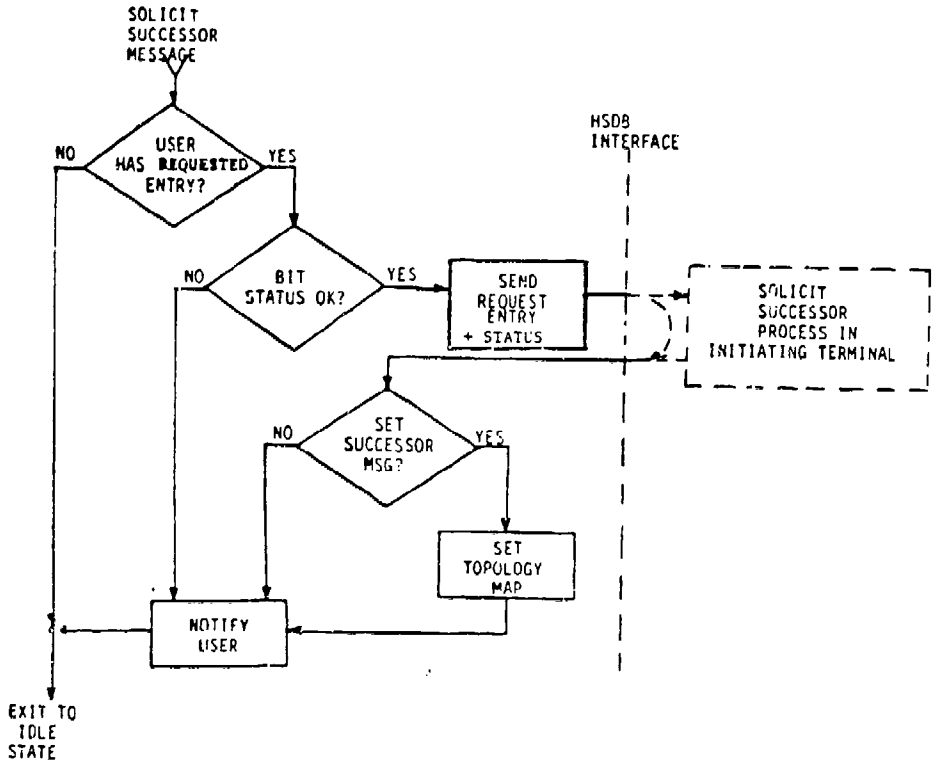


FIGURE 13
ENTER NET CONTROL FLOW DIAGRAM

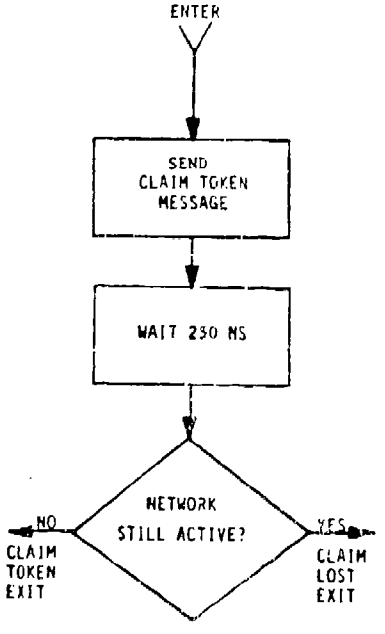


FIGURE 14
CLAIM TOKEN CONTROL FLOW DIAGRAM

Each terminal, upon recognizing the need to reinitialize the network, shall send a sequence of concatenated CLAIM_TOKEN messages. The number of messages shall be proportional to the terminal's TA according to the algorithm: "Quantity = TA+1 (1 to 128)". Each message shall consist of the sequence SD + FC + DA + $\overline{DA}$ + SA + FC + $\overline{FC}$ + FCS + ED.

When the CLAIM_TOKEN sequence is complete, the terminal shall wait 250 ns to allow for differential propagation time and shall then monitor the network for activity. If the bus is active, the terminal shall assume it has lost the CLAIM_TOKEN process and shall transition to IDLE state in order to allow another terminal to initiate COLD_START. If the network is quiet, it shall assume that it has won the claim process and shall initiate COLD_START.

A terminal listed in the topology memory as not operational on the "A" bus shall not initiate CLAIM_TOKEN state.

3.2.1.4.4.1.7 SOLICIT_ENTRY Process

The SOLICIT_ENTRY process shall be initiated from the PASS TOKEN state as described above. Figure 15 describes the process. Table XV describes the transaction sequence between soliciting terminal and the terminal desiring entry to the network.

a. The soliciting terminal shall send a SOLICIT_ENTRY message addressed to Potential Successor (PS).

b. The soliciting terminal shall wait RWT for a response from the network. If no response is recognized or if the response is not a REQUEST_ENTRY message then the terminal assumes that the request has had no response and prepares for the next SOLICIT_ENTRY process (step e).

c. If the request received a valid response, the terminal shall break the logical ring and splice PS into it by sending a SET_SUCCESSOR message to PS and a SET_PREDECESSOR message to CS.

d. The soliciting terminal shall update its own topology memory to include the new token rotation sequence and the status information from the new terminal and shall broadcast the new topology to the network via a SET_TOPOLOGY message.

e. The soliciting terminal shall prepare itself for the next SOLICIT_ENTRY process by incrementing PS (PS = PS + 1) and checking to determine if PS = CS. If PS = CS then reset PS to PS = TA + 1 to restart the cycle. If PS $\neq$ CS then retain PS.

f. The soliciting terminal shall reset its SET and transition to PASS_TOKEN state.

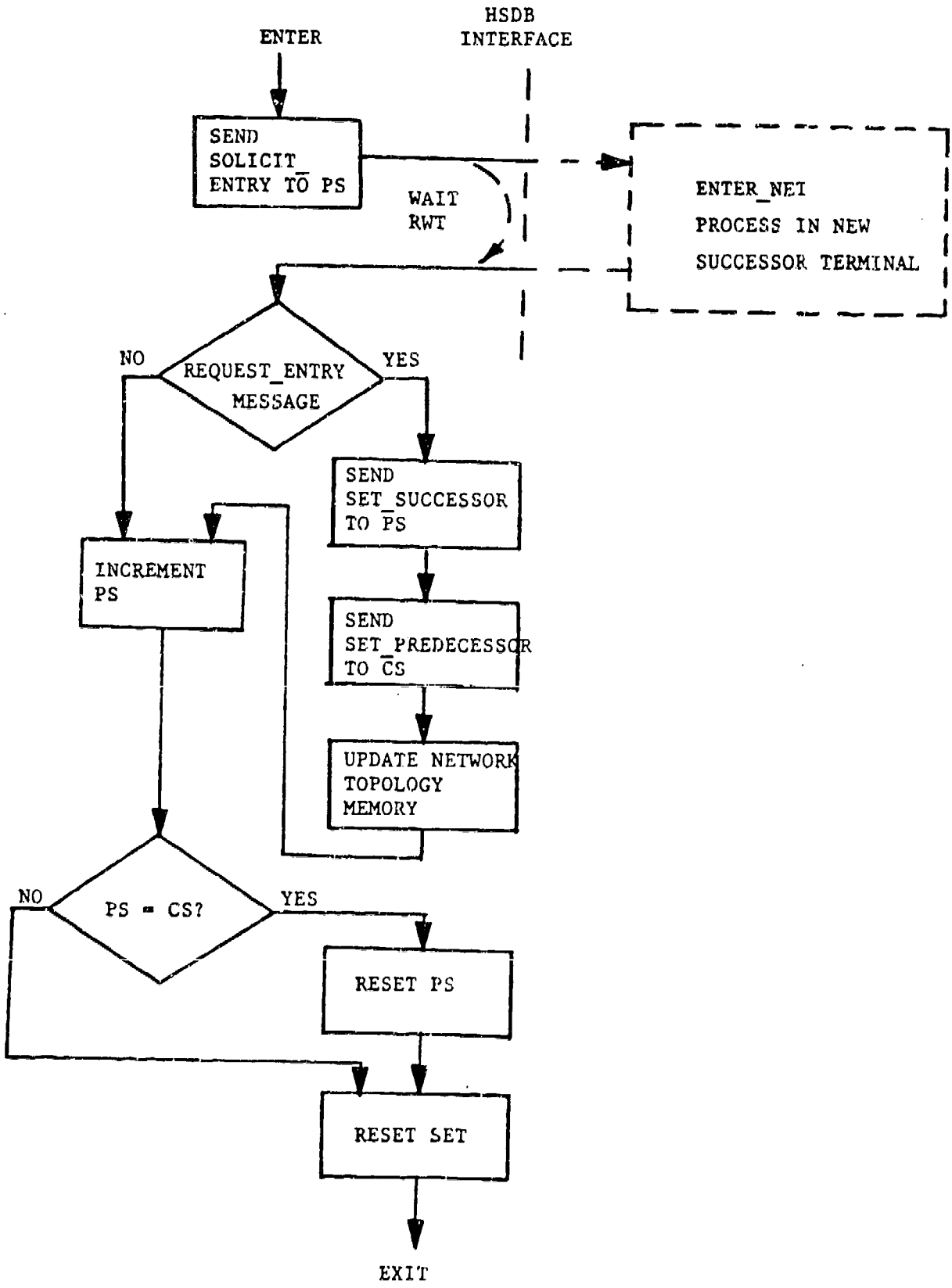


FIGURE 15
SOLICIT_ENTRY CONTROL FLOW DIAGRAM

3.2.1.4.4.1.8 LEAVE_NET Process

The LEAVE_NET process shall be initiated upon command of the terminal user or upon command from the network via a maintenance message. The terminal shall signal its intention to leave the network by issuing the following sequence of messages:

a. It shall broadcast a SET_TOPOLOGY message to the network splicing itself from the logical ring.

b. It shall pass the token to CS via a PASS_TOKEN_LEAVE_NET message.

c. It shall transition to the OFF_LINE state after monitoring to assure that CS has picked up the token.

The sequence of transmissions shall be accommodated using a concatenated message structure, using a single token hold cycle. It is a terminal design objective to provide sufficient locally stored power to complete the LEAVE_NET process following a power fault.

3.2.1.4.4.1.9 COLD_START Process

The COLD_START process shall be initiated from the CLAIM_TOKEN state whenever the token has been claimed and the topology memory is in its initial state. The terminal winning the token shall assume that it holds the highest address of any active terminal on the network and declares itself to be the ring master terminal. The ring master shall then proceed to establish the logical ring as illustrated by Figure 16.

a. The ring master shall define current successor (CS) as its own address (TA). It shall set its current predecessor (CP) and its potential predecessor (PP) counter to TA-1.

b. The ring master shall send a SOLICIT_ENTRY message to PP and wait for a period defined by the RWT for a response.

c. If a response is received in the form of a REQUEST_ENTRY message, the ring master shall respond with a SET_SUCCESSOR message addressed to PP listing CS as successor.

d. The ring master shall send a SET_PREDECESSOR message to CS listing PP as predecessor.

e. The ring master shall replace CS with PP and shall decrement PP (PP = PP-1).

f. Repeat b through e until PP<0.

g. The ring master shall send a SET_PREDECESSOR message to CS listing TA as predecessor.

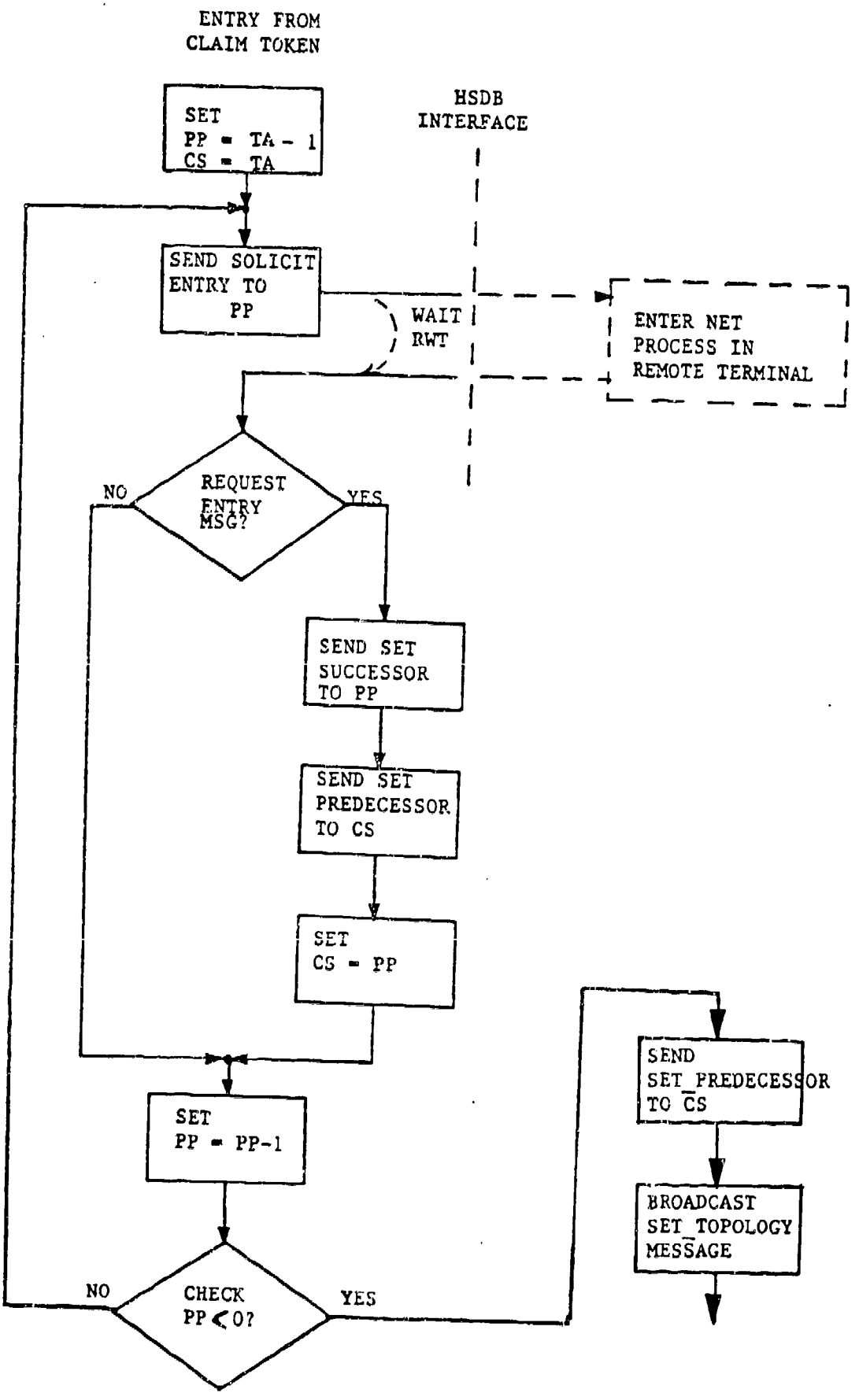


FIGURE 16

COLD_START CONTROL FLOW DIAGRAM

h. The ringmaster shall broadcast a SET_TOPOLOGY message including the physical address and status of each terminal residing on the network.

i. The ring master shall transition to the USE_TOKEN state.

3.2.1.4.4.1.10 RESTART Process

The RESTART process shall be initiated from the CLAIM_TOKEN state whenever the token has been claimed and the topology memory in the ring master terminal is intact. (Token lost and not recovered condition.) The following process shall be initiated:

a. The ring master shall poll each terminal listed as ON-LINE or OFF_LINE in its topology memory using a SOLICIT_REENTRY token message.

b. The ring master shall update its topology memory using the results of the poll. Any terminal responding with a REQUEST_REENTRY token message, shall be assumed to be present and active. Terminals not responding with a valid response message shall be assumed OFFLINE.

c. The ring master, at the conclusion of the poll, shall validate, and revise if necessary, the logical ring by splicing out any OFFLINE terminals within its topology memory.

d. The ring master shall broadcast a SET_TOPOLOGY message (if required) to update the topology memory of all other terminals.

e. The ring master shall transition to the USE_TOKEN state to re-initiate normal network operation.

3.2.1.4.4.1.11 Realtime Clock Management Process

The realtime clock management process shall be responsible for coordinating the selection of one terminal to act as network timekeeper and for synchronization and drift rate correction of all other realtime clocks to the timekeeper. Any terminal within the network may be designated by its user process as the timekeeper. In the absence of a designated timekeeper, one of the terminals shall assume the timekeeper function. The process is illustrated by figure 17.

a. Assuming that the terminal has been initialized as a not-timekeeper, the RST shall be initialized to TKF.

b. RST shall be tested to see if it has expired. If not, the process shall enter a wait loop until either RST expires (error case) or a SET_RTC message is received from the network (normal case).

c. Whenever a SET_RTC message is received from the network, the RTC shall be set to not-timekeeper mode of

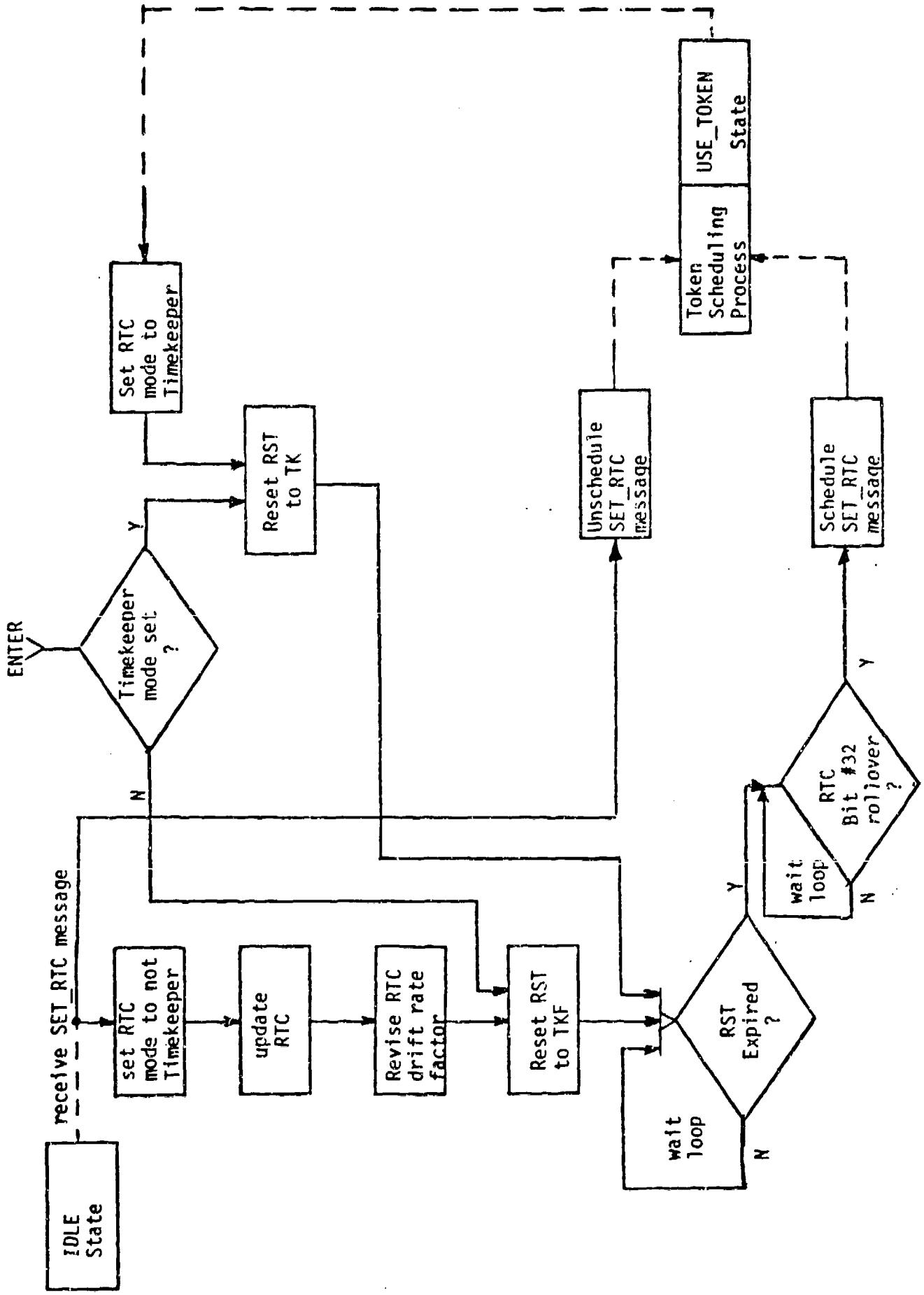


FIGURE (17)
REALTIME CLOCK MANAGEMENT PROCESS
CONTROL FLOW DIAGRAM

operation, the RTC shall be updated from the SET_RTC data field, and the RTC drift rate correction factor (KRT) shall be updated. RST shall be reset to TKF. The process shall test RST as described in section b, above.

d. Whenever the RST expires, the RTC shall assume that no other terminal is functioning as timekeeper and shall tentatively assume that function. The RTC shall be tested to synchronize with the FFFF to 0000 rollover (65ms) and then a SET_RTC message shall be scheduled. If a SET_RTC message is received from the network prior to the message being sent, the process shall revert to not- timekeeper mode of operation as described above.

e. If the SET_RTC message is sent, the RTC shall set itself to timekeeper mode of operation and shall proceed to generate SET_RTC messages at a rate defined by the initalization value TK.

The drift rate correction algorithm shall be implemented as an autonomous function of each terminal using the algorithm:

$$KRT_{new} = KRT_{old} + 1/2\ (KRT_{old} - KRT_{new})$$

Differential delay correction shall be accomplished by the user process when required to meet system RTC accuracy and synchronization requirements.

3.2.1.4.4.1.12 Redundant Bus Protocol

Terminals configured for operation using dual redundant buses, shall meet the following requirements in addition to those of preceding paragraphs. The general approach to redundant operation shall be for terminals to transmit and receive on both the "A" bus and the "B" bus at all times. The first message to arrive at the terminal ("early" bus) shall be used unless a data error occurs. In that case, the redundant message recovered from the "late" bus shall be used.

a. Transmit Operation. Normally a terminal shall transmit both messages and tokens on both the "A" bus and the "B" bus. The terminal shall modify that general rule whenever a transmitter failure has occurred on one of the output ports on an earlier transmission which has not been cleared by BIT. In this case, the terminal shall transmit on the non-error port alone.

b. Receive Operation. The terminal shall monitor both RX ports from the IDLE state. Messages shall be recovered, buffered, and validated from both the "A" bus and the "B" bus. The terminal shall accept the message from the "early" bus unless a data error is detected. In that case, the message from the "late" bus shall be accepted if valid. The decision shall be independently made

after receipt of each message sub-block ending in an FCS. Receiver errors shall be separately logged for each port.

3.2.1.4.4.2 User Interface Machine (UIM) Description

The UIM shall act as the intermediary between the user process and the other elements of the MAC layer. Its internal operation is not described in detail in this specification since it depends in large part on the characteristics of the specific user process being interfaced and because the MAC must function in accordance with the requirements of this specification no matter what method of implementing the UIM is elected. The UIM shall provide three different modes of operation, initialization, steady state operation, and maintenance.

During initialization, while the terminal is still in the OFF LINE state, terminal operating parameters may be loaded into the UIM via the TMU function. Parameters shall include:

a. Network topology memory
b. RWT initial value
c. THT initial value
d. TRT-P1 initial value
e. TRT-P2 initial value
f. TRT-P3 initial value
g. RMT initial value
h. SET initial value
i. RLT initial value
k. NIT initial value
l. RST initial value

As an option, the system/terminal designer may elect to use preset operating parameters. In this case, the logical ring may be established without initiation by the user. Once the terminal has entered the logical ring, the values of the operating parameters may be changed via a terminal maintenance message received from the network or by the user.

During steady state operation, the function of the UIM is to facilitate the transfer of data into and out of the terminal. This shall be accomplished in accordance with the requirements of paragraph 3.1.5.1.

Maintenance operations shall be initiated at the discretion of the user. Terminal operating parameters may be verified and/or modified. The terminal may be polled for statistics related to operation of the terminal or the network. As a minimum, the characteristics described in Table XI shall be computed within the UIM of each terminal.

It shall be possible to request the identical maintenance characteristics from any remote terminal in the network by transmission of a properly formatted terminal maintenance message. Similarly, it shall be possible to command any remote terminal in the network to

retransmit maintenance loopback messages back to the requesting terminal.

3.2.1.4.4.3 Receive Machine Description

The receive (RX) machine shall accept RXM from the PHY layer and shall generate data structures and signals for the MAC layer APM and UIM. Figure 18 describes the process which shall be implemented by the Rx machine. Note that for dual redundant terminals the process is duplicated for both the "A" bus and the "B" bus. The message from the "early" bus shall be accepted unless FCS validation fails.

The following approach is suggested in order to minimize terminal response time. Alternative approaches are allowed so long as all performance requirements are met.

The RXM is decomposed into its component fields as it is received. Appropriate responses are buffered in both ports of the Rx machine prior to the complete message being clocked into the machine. Action is deferred until the final bit has been received and the FCS has been validated. At this point, the decision as to whether to use the "A" buffers of the "B" buffers is made on the basis of FCS validation. The "early" message has precedence. The FCS is reset immediately following receipt of the last bit of each data block so that successive 256 word data blocks may be tested independent of each other.

3.2.1.4.4.4 Transmit Machine Description

The transmit (Tx) machine shall accept TXC and TXD from the ACM and UIM respectively and shall generate TXM for presentation to the PHY layer. TXU shall be sent first followed by TXD. FCS shall be appended to the end of TXD to form the composite TXM. For data messages greater than 256 words, the message shall be decomposed into a number of 256 word or remainder fields and shall be transmitted with a FCS field appended to each.

Figure 19 describes a suggested approach to design of the Tx machine. The approach is suggested as appropriate in order to minimize terminal response time. Alternative approaches are allowed so long as all performance requirements are met.

Transmission is initiated via TXC whenever the terminal receives a message requiring a transmission response. The APM assumes that FCS will be validated and begins to transmit PRE + SD. If FCS does not validate, the ABORT sequence is transmitted. If the FCS does validate, the Tx machine continues the transmission by sending the scheduled FC + DA + SA + WC + data + FCS sequence. FCS is generated for each 256 words and appended to the data field in realtime.

3.2.1.5 Error Recognition and Recovery

The HSDB shall include design features which limit the impact of failure mechanisms. No single failure shall be capable of disabling the entire network. Each terminal shall have the capability to

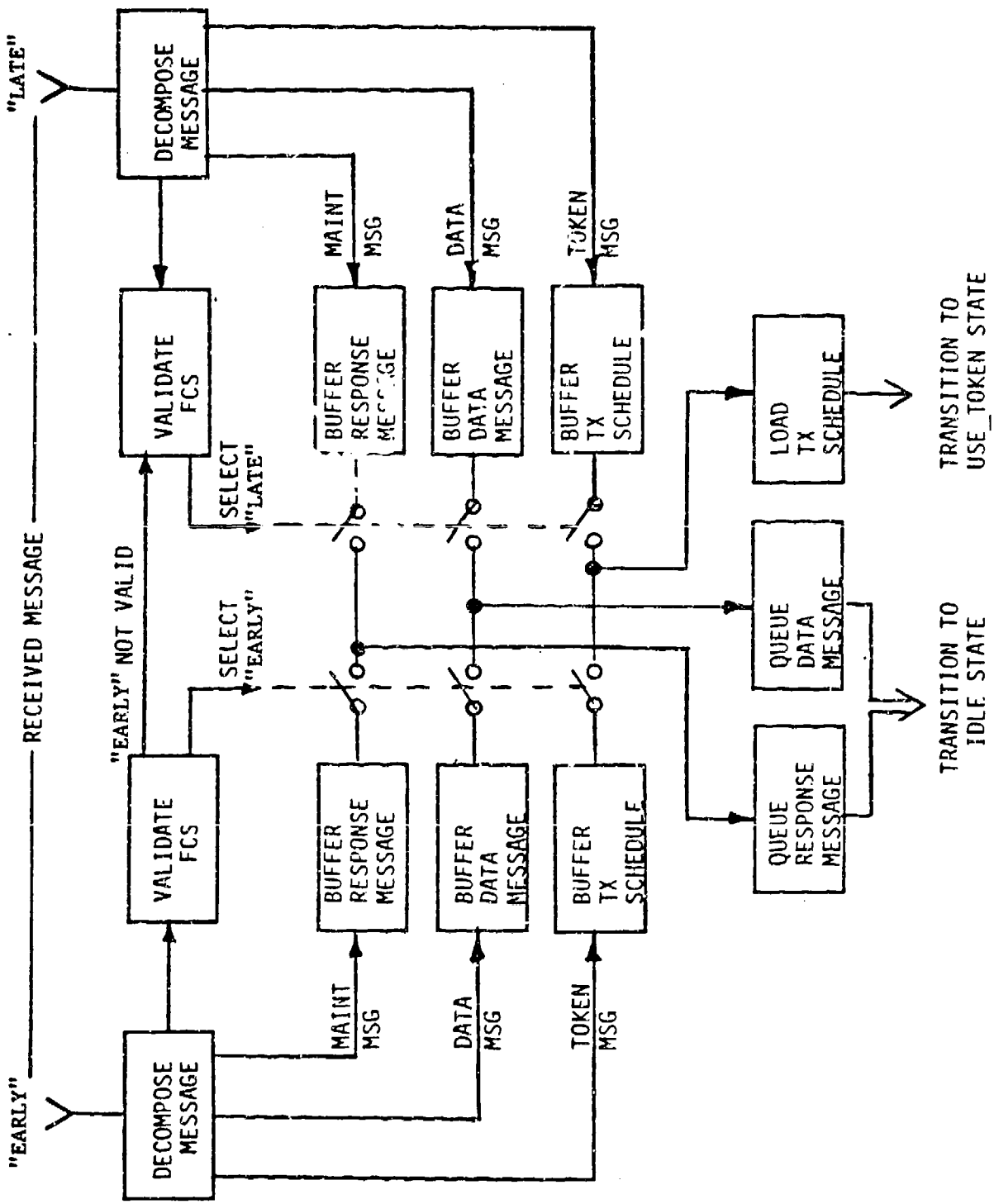


FIGURE 18
SIMPLIFIED RECEIVE MACHINE
FUNCTIONAL FLOW DIAGRAM

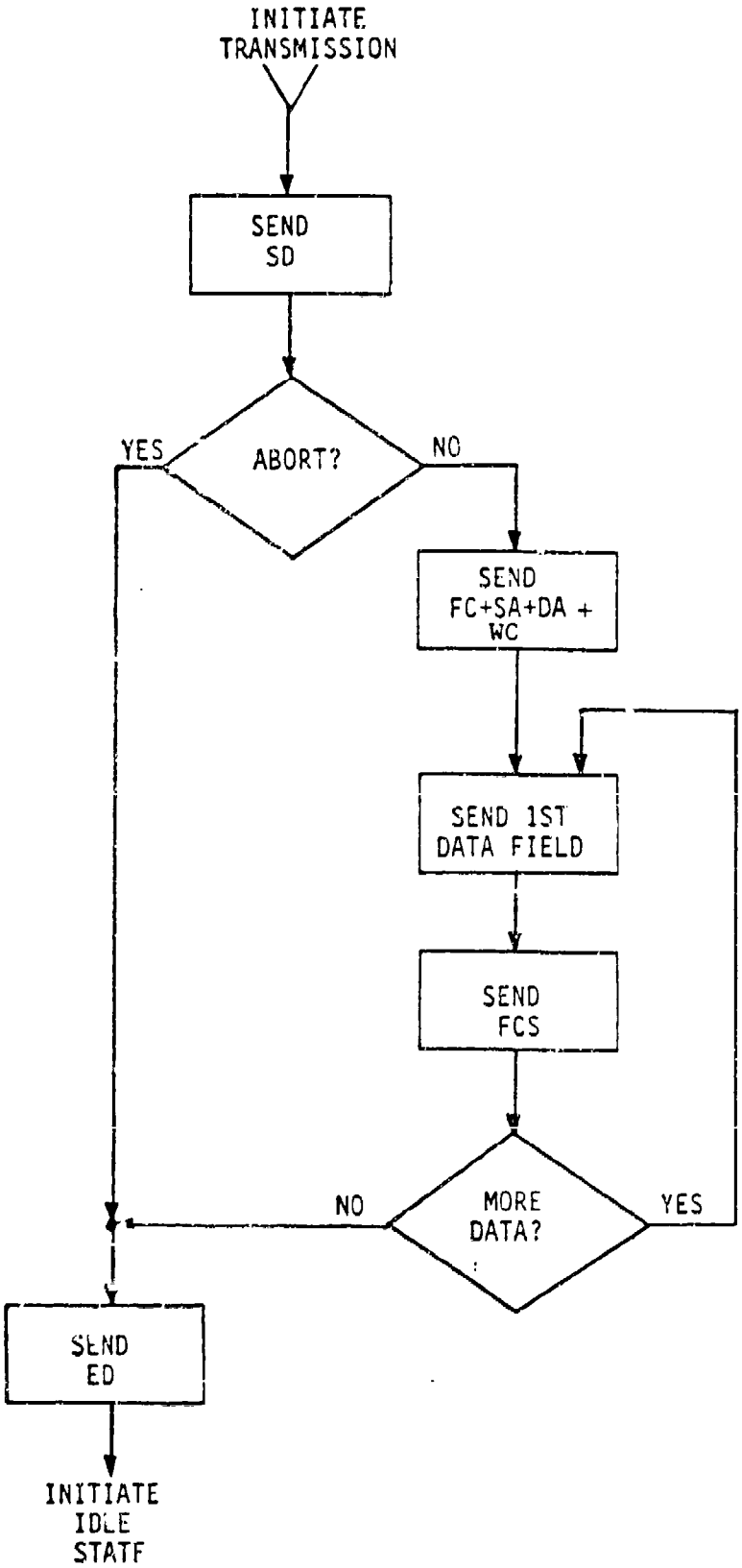


FIGURE 19
SIMPLIFIED TRANSMIT MACHINE
FUNCTIONAL FLOW DIAGRAM

recognize all errors defined by Table XVII and to recover from the error state in the manner described. The general error handling approach shall be to treat received messages which do not meet all quality tests as noise; the sending terminal shall be responsible for recovery from a lost token or multiple token condition; and each terminal shall contain embedded features which prevent long term corruption of the bus.

TABLE XVII
ERROR RECOGNITION AND RECOVERY

Error Condition	Recognition Strategy	Recovery Strategy
Message date field corrupted on bus	Receiver Data error	Message discarded
Message DA field corrupted on bus	Receiver Data error	Message discarded
Message FC field corrupted on bus	Receiver Data error	Message discarded
Message token field corrupted on bus	Token lost error	Recover token Process
Message preamble corrupted on bus	Receiver sync error	Message lost
Message SD field corrupted on bus	Receiver framing error	Message lost
Message ED field corrupted on bus	Receiver WC and framing error	Message accepted
Message SA field corrupted on bus	Receiver Data error	Message discarded
Terminal failure - not holding token	Token lost error	Recover token Process
Terminal failure - holding token	Bus activity error	WARM_RESTART Process
Terminal transmitter stuck 'on'	Token hold error	WARM_RESTART Process
Terminal receiver defective	Net initialize error	Terminal goes off-line
Terminal transmitter defective	Transmitter error	Terminal goes off-line
Terminal clock high or low freq	Receiver sync error	Terminal goes off-line
Terminal receiver low sensitivity	Receiver error	Receive only operation
Terminal transmitter low output	Transmitter error	Receiver only operation
Terminal address duplicate	Token error (multiple)	Message discarded
Terminal address intermittent	Multiple/lost token error	Retry
All/most terminals off-line concurrently	Bus activity error	COLD_RESTART process
Bus broken - redundant bus topology	Bus activity error	SWITCH_BUS process
Bus broken - single bus topology	Bus activity error	WARM/COLD_RESTART process

3.2.1.5.1 Recognition Strategies

The general strategy for error recognition shall be for each terminal to use all information available from past network activity to infer the health state of itself and of the network in total. Error counters included in each terminal shall accumulate detected errors for maintenance status reporting.

3.2.1.5.1.1 Receiver Data Error

The receiver data test function shall signal if FCS or Manchester bit errors occurred between the SD and the ED fields of the received message. In all cases, the message shall be discarded by the receiving terminal. The data error counter shall be incremented for each detected FCS error or Manchester error.

3.2.1.5.1.2 Token Lost Error

In the case where a discarded message contained the token, the sending terminal shall recognize that the token was lost and shall initiate the recovery procedure. The token lost error counter shall be incremented for each detected token lost error.

3.2.1.5.1.3 Receiver Synchronization Error

The receiver clock synchronization test function shall signal if the terminal clock failed to synchronize to the received message or lost synchronization during receipt of the message. In all cases, the message shall be discarded by the receiving terminal. The receive sync error counter shall be incremented for each detected sync error.

3.2.1.5.1.4 Receiver Framing Error

The receiver framing error test function shall signal if the terminal received a message which was not properly framed by SD and ED fields. If all other quality tests pass, the message shall be accepted as valid by the terminal. In all cases, the framing error counter shall be incremented for each detected framing error.

3.2.1.5.1.5 Word Count Error

The receiver word count error test function shall signal if the terminal received a message in which the value in the word count field did not match the number of words in the data field. The word count error counter shall be incremented for each detected word count error.

3.2.1.5.1.6 Deleted

3.2.1.5.1.7 Bus Activity Error

Whenever the NIT expires, the receiver shall signal a bus activity error and shall increment the bus activity error counter.

3.2.1.5.1.8 Token Hold Error

A token hold error shall be signaled if the terminal receiver function detects that the transmitter has been active for a period greater than the THT allows or if the watchdog timer expires. In the former case, the terminal shall terminate the transmission with an abort sequence as described in 3.2.1.1. In either case, the terminal shall initiate an error shutdown sequence and shall go off-line. The token hold error counter shall be incremented at each occurrence of a token hold error.

3.2.1.5.1.9 Network Initialize Error

The network initialize error state shall be recognized whenever the terminal attempts to establish the logical ring via the initialization process but fails to find a successor terminal. The terminal shall assume that either its receiver is inactive or else it is alone on the network and shall go off-line. The network initialize error counter shall be incremented at each occurrence of initialize error.

3.2.1.5.1.10 Transmitter Error

A transmitter error shall be signaled whenever the terminal receiver does not recover a loopback signal of expected amplitude and with correctly encoded data from its companion transmitter and is not in a receiver error state. The terminal shall set that transmitter port to "inactive" status and shall record that fact in its topology memory. Should that port be the only active transmit port of the terminal, it shall abort the transmission with a valid abort sequence and shall go off-line. Should an active transmit port remain, the terminal shall enter a TOPOLOGY_MESSAGE into its transmit queue to affect a broadcast of its change in status and shall continue its present TOKEN_HOLD cycle as if the error had not occurred. The transmitter error counter shall be incremented at each occurrence of transmitter error.

3.2.1.5.1.11 Receiver Error

A receiver error shall be signaled whenever the terminal receiver fails to recover three successive packets from the network which are error free (Manchester error, FCS error, or framing error). Should a receive error state exist, the terminal shall set that receive port to inactive status and shall record that fact in its topology memory. It shall also enter a TOPOLOGY_MESSAGE into its transmit queue to affect a broadcast of its change in status. Should that port be the only active receive port of the terminal, it shall go off-line. The receive error counter shall be incremented at each occurrence of receiver error.

3.2.1.5.1.12 Token Error

If a terminal receives a token from the network with the SA different from that listed as its predecessor in the topology memory, the terminal shall assume that a multiple token exists and shall not

respond. The token error counter shall be incremented at each occurrence of a multiple token error. The terminal shall retain its position in the logical ring.

If a terminal recognizes a token on the network with SA = its PA and DA ≠ TA, it shall assume that its topology memory is defective and shall not transition to USE_TOKEN state. It shall enter a REPORT_TOPOLOGY_MEMORY message addressed to the ring master terminal. The token error counter shall be incremented at each occurrence.

3.2.1.5.2 Recovery Strategy

HSDB terminals shall support a hierarchy of recovery processes.

1st - Alternate bus
2nd - Retry
3rd - Recover token
4th - Restart

The design objective is to cause a minimum disruption to the network under failure/error conditions. Messages which are corrupted on the network and received as defective shall be abandoned (lost). The terminal protocol shall contain features which recover from lost token messages. The user shall be responsible for determining which messages are critical to system operation and for instituting recovery of lost data messages and maintenance messages except in instances specifically defined elsewhere in this specification.

3.2.1.5.2.1 Alternate Bus Procedure

The alternate bus shall be used as the first attempt at recovery of a lost message. Each terminal shall monitor both RX ports. Messages shall be recovered and buffered from both the "A" bus port and the "B" bus port. Each terminal shall accept the message from the earliest arriving message ("early" message) unless a data error is detected. In the event of a data error on the "early" bus port, the message from the other ("late") port shall be accepted unless it too contained a data error.

3.2.1.5.2.2 Retry Procedure

Retry shall be automatically initiated for lost token messages only. Each terminal shall contain design features which implement the following retry strategy:

a. The token sending terminal shall monitor the network for a period of time immediately following its transmission of a token message.

b. If after RWT, the sending terminal does not recognize the pattern "PRE + SD + FC" on the network, it shall assume that the token has been lost during transmission and shall retry the token pass.

c. The sending terminal shall retry a maximum of RLT times to pass the token. If not successful, the terminal shall assume that the intended successor terminal has failed and shall institute token recovery procedures.

3.2.1.5.2.3 Recover Token Procedure

The recover token procedure shall be automatically initiated whenever (A) the intended successor terminal has failed prior to accepting the token, or (B) the token holding terminal has failed while in possession of the token. Each terminal shall contain design features which implement the following token recovery strategy.

Case A: Case A shall be initiated by the token holding terminal upon unsuccessful completion of the retry strategy. Four activities shall be accomplished in the sequence defined below.

a. The terminal shall modify its topology memory to splice around its present (failed) successor.

b. The terminal shall broadcast the changed topology memory to the network via a SET_TOPOLOGY_MEMORY message.

c. The terminal shall pass the token to its new successor.

d. The terminal shall attempt to recover the token a maximum of RLT times. If not successful, the terminal shall assume that its own receiver has failed and shall transition to the off-line state.

Case B: Case B shall be initiated by the predecessor terminal of the terminal which held (lost) the token. Each terminal shall monitor the entire token hold cycle of its successor to ensure that the token is passed (to successor's successor). Five activities shall be accomplished in the sequence defined below.

a. If the last message transmitted as part of successors packet was not recognized as a token message, the terminal shall continue to monitor the network for RWT.

b. If after RWT the terminal does not recognize the pattern "PRE + SD + FC" on the network, it shall assume that the token holder has failed.

c. The terminal shall modify its topology memory to splice around its present (failed) successor.

d. The terminal shall broadcast the changed topology memory to the network via a SET_TOPOLOGY_MEMORY message.

e. The terminal shall pass the token to its new successor.

If the terminal recognizes valid network activity during RWT, it shall assume that its own receiver lost the token message and shall remain quiescent.

3.2.1.5.2.4 Restart

Restart shall be automatic whenever the token has been lost and is not recoverable by the alternate bus, retry, or recover token procedures. Each terminal shall contain design features in accordance with the restart requirements of 3.2.1.4.4.1.10.

3.2.2 Physical Characteristics

Physical characteristics shall be defined by specifications associated with specific systems applications and specific hardware components.

3.2.3 Reliability (No requirement)

3.2.4 Maintainability (No requirement)

3.2.5 Availability (No requirement)

3.2.6 System Effectiveness Models (No requirement)

3.2.7 Environmental Conditions

The HSDB and components thereof shall be designed to withstand the extremes of a MIL-E-5400, Class II temperature environment without sustaining damage or degraded operation.

3.2.8 Nuclear Control Requirements (No requirement)

3.2.9 Transportability (No requirement)

3.3 Design and Construction

MIL-E-5400T shall be used as a guideline for the design of HSDB hardware components.

3.3.1 Materials, Processes, and Parts

HSDB hardware components shall be designed using materials, processes, and parts selected to be in accordance with the requirements of MIL-E-5400T, paragraph 3.1 through 3.1.58 to the maximum extent practical.

3.3.2 Electromagnetic Radiation (No requirement)

3.3.3 Nameplates and Product Markings

HSDB assemblies shall be individually identified and marked with an unique Collins parts number and serial number.

3.3.4 Workmanship

Workmanship shall be inspected to be in general accordance with the requirements of MIL-E-5400T, paragraph 3.5.

3.3.5 Interchangeability

Wherever practical, like part numbered assemblies and subassemblies shall be interchangeable mechanically and electrically without adjustment of that or other system components.

3.3.6 Safety

The HSDB and components thereof shall be designed using the requirements of MIL-E-5400T, paragraph 3.2.22 as a guideline.

3.3.7 Human Performance/Human Engineering (No requirement)

3.4 Documentation

Documentation of the HSDB system and components thereof shall conform to the requirements of DoD-D-1000.

3.5 Logistics (No requirement)

3.6 Personnel and Training (No requirement)

3.7 Functional Area Characteristics

A coaxial HSDB system shall be comprised of coax bus terminal, coupler, mainbus, stub bus, and termination assemblies as illustrated by figure 20a. A FO HSDB system shall be comprised of a star coupler, FO bus terminal, attenuator, and fiber functions as illustrated by figure 20b.

3.7.1 Coaxial Bus Function Descriptions

3.7.1.1 Coax Bus Terminal

The coax bus terminal function is responsible for establishing communication paths between user functions via the HSDB network and for organization of data for transport between users. The terminal function shall be designed using a combination of hardware and software subfunctions to perform as described by paragraph 3.2 of this specification. The interface with the user process shall meet the requirements of paragraph 3.1.5.1. The interface with the coupler shall meet the requirements of 3.7.5.2.

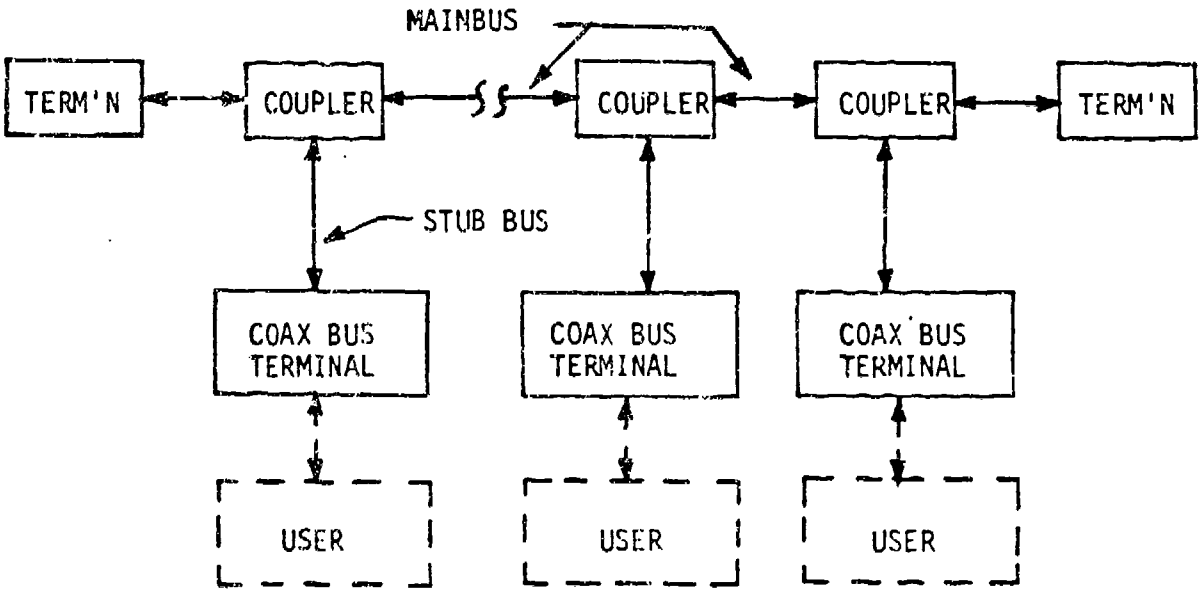


FIGURE 20a

COAXIAL HSDB SYSTEM FUNCTIONAL PARTITIONING

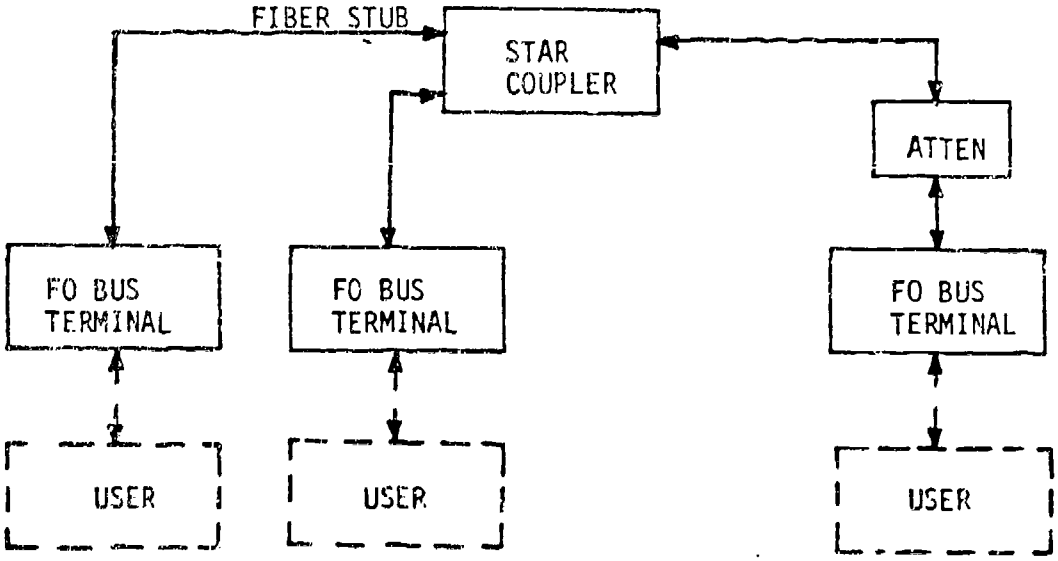


FIGURE 20b

FO HSDB SYSTEM FUNCTIONAL PARTITIONING

3.7.1.2 Coupler

The coax bus coupler function allows physical and electrical access to the bus media (coaxial cable) by the terminal. The coupler function shall be comprised of a hardware assembly meeting the requirements of paragraph 3.2.1.2.1-e through 3.2.1.2.1-1. The interface with the terminal function shall meet the requirements of 3.1.5.2.1, 3.1.5.2.2, and 3.1.5.2.3. The interface with the mainbus shall meet the requirements of 3.1.5.2.4 (2 ports, input and output).

3.7.1.3 Mainbus

The mainbus function allows couplers (and terminals) to be physically separated by a maximum of 300 feet. The mainbus shall be comprised of a set of coaxial cable assemblies meeting the requirements 3.2.1.2.1-e, and 3.2.1.2.1-f (less coupler mainbus loss). The interfaces with the couplers shall meet the requirements of 3.1.5.2.4.

3.7.1.4 Stub Bus

The stub bus function allows terminals to be physically separated from couplers by a maximum of 20 feet. The stub bus shall be comprised of a set of coaxial cable assemblies meeting the requirements 3.2.1.2.1-a through 3.2.1.2.1-d. The interfaces shall meet the requirements of 3.1.5.2.1 through 3.1.5.2.3.

3.7.1.5 Termination

The termination function is used at each physical extremity of the network to provide an ideal impedance at the unused mainbus port of the last coupler. The characteristic impedance shall be 50+J0 ohms ±TBS ohms. The interface shall consist of a male type TNC connector.

3.7.2 Fiber Optic Bus Function Descriptions

3.7.2.1 Fiber Optic Bus Terminal

The fiber optic (FO) bus terminal function is responsible for establishing communications paths between user functions via the HSDB network and for organization of data for transport between users. The terminal function shall be designed using a combination of hardware and software subfunctions to perform as described by paragraph 3.2 of this specification. The interface with the user process shall meet the requirements of 3.1.5.1. The interface with the star coupler shall meet the requirements of 3.1.5.3.

3.7.2.2 Star Coupler

The star coupler function allows optical interconnection of the terminals comprising the network. The coupler shall be comprised of a single hardware assembly meeting the requirements of 3.2.1.2.2-b through 3.2.1.2.2-f. The interface shall meet the requirements of 3.1.5.3 (each of 64 ports).

3.7.2.3 Fiber Stub

The fiber stub function allows terminals to be physically separated from the star coupler by a maximum of 300 feet. The fiber stub shall be comprised of a set of fiber optic cables terminated at each end by either a connector or a splice, and shall meet the requirements of 3.2.1.2.2-a and 3.2.1.2.2-c through 3.2.1.2.2-f. The interface at the terminal end shall meet the requirements of 3.1.5.3.

3.7.2.4 Attenuator

The attenuator function allows terminals in low loss paths of the network to avoid saturation of the terminal receiver by lowering the optical power prior to application to the terminal. The attenuation shall be specified by the system designer at a value appropriate for this purpose given the specific characteristics of the network under development. Interface with the fiber stub shall be via splice or optical SMA type connectors at the discretion of the system designer.

3.8 Precedence

Not Applicable

4. QUALITY ASSURANCE PROVISIONS

Requirements for formal tests/verifications of system performance design characteristics shall be specified by the development specifications of individual systems and system components.

5. PREPARATIONS FOR DELIVERY (Not required)

6. NOTES

6.1. The user interface to the terminal is purposely specified in a general manner so as to allow interface to a variety of different users in an optimum manner. Some interfaces may be implemented using serial data paths, others using parallel, for example. The development specification of each terminal/user must specify this interface in detail.

10. GLOSSARY OF ACRONYMS

Acronym	Description	Definition
APM	Access and Protocol Machine	3.2.1.4.3.1
CRU	Configuration Reporting Unit	3.2.1.4.1.11
CS	Current Successor Address	3.2.1.4.4.1.9
CP	Current Predecessor Address	3.2.1.4.4.1.9
DA	Destination Address	3.2.1.1.5
ED	End Delimiter	3.2.1.1.9
F1	TRT_P1 Flag	3.2.1.4.4.1.3
F2	TRT_P2 Flag	3.2.1.4.4.1.3
F3	TRT_P3 Flag	3.2.1.4.4.1.3
FC	Frame Control	3.2.1.1.3
FCS	Frame Check Sequence	3.2.1.1.8
FIFO	First-In-First-Out	
FO	Fiber Optic	
HSDB	High Speed Data Bus	
KRT	RTC Drift Rate Correction	3.2.1.4.4.1.11
LS	Logical Successor	
MAC	Media Access Control Layer	3.2.1.4
MED	Media Layer	3.2.1.2
NIT	Network Inactivity Timer	3.2.1.4.3.5
P0	Highest Priority	
P1	Intermediate Priority	
P2	Lowest Priority	
P3	No Priority	
PHY	Physical Layer	3.2.1.3
PP	Potential Predecessor	3.2.1.4.4.1.9
PRE	Preamble	3.2.1.1.1
PS	Potential Successor	3.2.1.4.4.1.7
RLT	Retry Limit Timer	3.2.1.4.3.7
RMT	Ring Maintenance Timer	3.2.1.4.3.3
RST	Realtime Clock Synchronism Timer	3.2.1.4.3.8
RTC	Realtime Clock	3.2.1.4.2.7
RWT	Response Window Timer	3.2.1.4.3.6
RX	Receive	
RXC	Receive Control Unit	3.2.1.4.1.9
RXD	Receive Data Unit	3.2.1.4.1.8
RXM	Receive Message	3.2.1.4.1.7
RXU	Receive Unit	3.1.5.1/3.2.1.4.1.10
SA	Source Address	3.2.1.1.4
SD	Start Delimiter	3.2.1.1.2
SET	Solicit Entry Timer	3.2.1.4.3.4
TA	Terminal Address	
TCU	Terminal Control Unit	3.2.1.4.1.5
THT	Token Hold Timer	3.2.1.4.3.1
TK	Timekeeper	3.2.1.4.3.8
TKF	Not Timekeeper	3.2.1.4.3.8
TMU	Terminal Management Unit	3.1.5.1/3.2.1.4.1.2
TRT	Token Rotation Timer	3.2.1.4.3.2
TRU	Transmit/Receive Unit	3.2.1.3.n
TSU	Terminal Status Unit	3.1.5.1/3.2.1.4.1.12
TX	Transmit	
TXC	Transmit Control Unit	3.2.1.4.1.4

10. GLOSSARY OF ACRONYMS (Cont'd)

Acronym	Description	Definition
TXD	Transmit Data Unit	3.2.1.4.1.3
TXM	Transmit Message	3.2.1.4.1.6
TXU	Transmit Unit	3.1.5.1/3.2.1.4.1.1
UIM	User Interface Machine	3.2.1.4.3.2
WC	Word Count	3.2.1.1.6

20 PI-Bus Interface Design Guide

20.1 Scope

This appendix to the HSDB System Specification illustrates an approach which may be used to interface the HSDB to a PI-Bus. The PI-Bus was chosen due to its potential for wide application throughout military aircraft as a backplane bus.

20.2 Requirements

The user interface requirements of a HSDB terminal are described in paragraph 3.1.5.1. PI-Bus interface requirements are described in the PI-Bus specification.

20.2.1 Transmit Operation

The transmit operation is defined as the process required for a PI-Bus module to place a message on the HSDB. The operation consists of the set of transactions described in Table I. The SEQuence column identifies the transaction sequence described in the HSDB System Specification. The SOURCE column identifies the PI-Bus module which wishes to originate the message. The DESTination column is the PI-Bus module which has direct interface (gateway) to the HSDB. MESSage refers to the message sequence type as described in the PI-Bus specification. Note that it is assumed that the SOURCE module is PI-Bus master at initiation of the operation.

TABLE I
TRANSMIT OPERATION

SEQ	SOURCE	DEST	MESS	NOTES
1(TMU)	*	-------------->	BUS INTERFACE MESSAGE	a.
2(TSU)	<--------------	*	PARAMETER WRITE	b.
3(TXU)	*	-------------->	BUS INTERFACE MESSAGE OR BLOCK MESSAGE	c.
4(TMU)	*	-------------->	PARAMETER WRITE	d.

NOTES:
- * PI-Bus Master
- --> Information Destination
- a. Request to send, word count, class of service, destination address and subaddress, priority
- b. Buffer address or error/busy indication
- c. Data field contents
- d. Confirmation or abort

20.2.2 Receive Operation

The receive operation is defined as the process required for a PI-Bus module to recover a message from the HSDB. The operation consists of the set of transactions described in Table II. The SEQuence column identified the transaction sequence described in the HSDB system specification. The SOURCE column identifies the

PI-Bus module which has direct access to the HSDB. The DESTination column is the PI-Bus module for which the message is intended. Note that it is assumed that the SOURCE module is PI-Bus master at initiation of the operation. The MESSage column refers to the message sequence type as described in the PI-Bus specification.

TABLE II
RECEIVE OPERATION

SEQ	SOURCE	DEST	MESS	NOTES
1(TSU)	*	-------------->	BUS INTERFACE MESSAGE	a.
2(RXU)	----------	>*	BUS INTERFACE MESSAGE OR BLOCK MESSAGE	b.
3(TMU)	<----------	*	PARAMETER WRITE	c.

NOTES:
- * PI-Bus Master
- --> Information destination
- a. Word count, location of buffer, source address, class of service, priority
- b. Data field contents
- c. Confirmation (free buffer)

20.2.3 Status Request Operation

The status request operation is defined as the process by which a PI-Bus module may request status information from either the local HSDB interface module or of some other terminal of the HSDB. The operation consists of the set of transactions listed on Table III. The SOURCE column identifies the PI-Bus module which originates the status request. The DEST column identifies the PI-Bus module having direct access to the HSDB.

TABLE III
STATUS REQUEST OPERATION

SEQ	SOURCE	DEST	MESS	NOTES
1(TMU)	*	-------------->	BUS INTERFACE MESSAGE	a.
2(TSU)	<----------	*	BUS INTERFACE MESSAGE	b.
3(RXU)	*<----------		BUS INTERFACE MESSAGE OR BLOCK MESSAGE	c.
4(TMU)	*	-------------->	PARAMETER WRITE	d.

NOTES:
- * PI-Bus master
- --> Information destination
- a. Status request, parameter(s), source address
- b. Word count, location of buffer, source, class of service
- c. Status message
- d. Confirmation (free buffer)

20.2.4 Status Report Operation

The Status Report Operation is defined as the process by which the PI-Bus module with direct access to the HSDB may initiate a status report to another PI-Bus module. The operation consists of the set of transactions listed in Table IV. The SOURCE column identifies the PI-Bus module which originates the status request (HSDB module).

TABLE IV
STATUS REPORT OPERATION

SEQ	SOURCE	DEST	MESS	NOTES
1(TSU)	*---------->		PARAMETER WRITE	a.

NOTES:
- * PI-Bus master
- --> Information destination
- a. Status summary

☆U.S.Government Printing Office: 1987 — 548-054/80514