

# DEMFA HARDWARE FUNCTIONAL SPECIFICATION

Order Number: XXX

This document specifies the functionality of the DEMFA, the XMI to FDDI adapter.

Revision

1.3

**digital**  
**DIGITAL INTERNAL USE ONLY**

**Written by:**  
The DEMFA Group

**Edited by:**  
Ron Edgar  
Santosh Hasani  
Lea Walton

VAX PRODUCTS AND OPTIONS  
14-March-1991

**Digital Equipment Corporation**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

---

**REVISION HISTORY**

---

Date	Revision	Summary of Changes
1/13/89	X00	First cut
1/20/89	X01	First 'complete' edition
3/7/89	X02	Expanded and corrected
5/5/89	X03	Further expanded and corrected - for group internal review
5/19/89	X04	Expanded and corrected as the result of group internal review. This edition for external review
2/7/90	X05	Registers updated
6/30/90	X06	LED table included, More complete Environment conditions mentioned, Some other overhauling
7/20/90	1.0	Major update of All Chapters
1/8/91	1.1	Further updates from DEMFA notesfiles, design reviews, and PMC2.
3/1/91	1.2	Change to PMC Register CSR03; protocol errors 24 and 25 are reversed.
3/14/91	1.3	Included DEMFA MTBF figures.

---

Copyright ©1988,1989 by Digital Equipment Corporation

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may occur in this document.

This specification does not describe any program or product which is currently available from Digital Equipment Corporation. Nor does Digital Equipment Corporation commit to implement this specification in any product or program. Digital Equipment Corporation makes no commitment that this document accurately describes any product it might ever make.

This document was prepared using VAX DOCUMENT, Version 1.2

# CONTENTS

<b>Chapter 1 DEMFA INTRODUCTION</b> .....	1
1.1 INTENDED AUDIENCE .....	1
1.2 GOALS AND NON-GOALS .....	1
1.3 ASSOCIATED DOCUMENTS .....	2
1.4 TERMINOLOGY .....	3
1.5 FDDI/802.2 Frame Format .....	4
<b>Chapter 2 DEMFA EXECUTIVE SUMMARY</b> .....	5
2.1 DEMFA CONTEXT .....	5
2.2 DEMFA OVERVIEW .....	6
2.2.1 HOST INTERFACE .....	7
2.2.2 THE ADAPTER MANAGER SUBSYSTEM .....	7
2.2.3 THE PACKET BUFFER MEMORY SUBSYSTEM .....	7
2.2.4 THE FDDI CHIPSET AND THE PARSER .....	8
2.3 PACKET RECEPTION .....	11
2.4 PACKET TRANSMISSION .....	11
2.5 PERFORMANCE .....	12
<b>Chapter 3 DEMFA FUNCTIONAL DESCRIPTION</b> .....	15
3.1 HOST INTERFACE .....	15
3.1.1 THE XMI CORNER .....	15
3.1.2 DEMFA SUPPORTED XMI TRANSACTIONS .....	15
3.1.3 XMI INTERFACE .....	17
3.1.3.1 THE XCI BUS .....	17
3.1.3.2 THE PBI BUS .....	18
3.1.3.2.1 PBI BUS PROTOCOL .....	18
3.1.3.3 THE ESP BUS .....	19
3.1.4 THE ESP GATE ARRAY .....	19
3.2 ADAPTER MANAGER SUBSYSTEM .....	22
3.2.1 OVERVIEW .....	22
3.2.2 AMI INTERFACES .....	23
3.2.2.1 THE ESP BUS .....	24
3.2.3 MICROPROCESSOR .....	25
3.2.4 AMI GATE ARRAY .....	26
3.2.4.1 FUNCTIONAL UNITS .....	27
3.2.5 68020 MEMORY SUBSYSTEM .....	29
3.2.6 SUBSYSTEM DATA INTEGRITY .....	29

3.2.7 POWER UP & SELF TEST .....	30
3.3 PMC GATE ARRAY .....	30
3.3.1 OVERVIEW .....	30
3.3.2 INTERFACES .....	32
3.3.2.1 RMC INTERFACE .....	32
3.3.2.2 ESP INTERFACE .....	32
3.3.3 FUNCTIONAL BLOCK DESCRIPTIONS .....	32
3.3.3.1 PACKET BUFFER MEMORY .....	32
3.3.3.2 PAGE TABLE AND BUFFER DESCRIPTOR MEMORY .....	32
3.3.3.3 RING ENTRY MOVER .....	33
3.3.3.4 PMT - Packet Memory Test .....	34
3.3.3.5 MIF - MEMORY CONTROLLER .....	34
3.3.3.6 RMC I/F .....	34
3.3.3.7 ESP I/F .....	34
3.3.4 TRANSMIT/RECEIVE DATA STREAM PRIORITY .....	35
3.3.5 ADDITIONAL FEATURES .....	35
3.4 THE FDDI CHIP SET .....	35
3.5 PARSER SUBSYSTEM .....	37
3.5.1 OVERVIEW .....	37
3.5.2 THE PARSER GA .....	42
3.5.2.1 RMC INTERFACE .....	42
3.5.2.2 ADAPTER MANAGER INTERFACE .....	42
3.5.2.3 RAM INTERFACE .....	42
3.5.2.4 PACKET MEMORY CONTROLLER INTERFACE .....	42
3.5.2.5 ALGORITHM STATE MACHINE .....	42
3.5.2.6 BUFFER DESCRIPTOR DECODE STATE MACHINE .....	42
3.5.2.7 SAP DECODE STATE MACHINE .....	43
3.5.2.8 TEST INTERFACE .....	43
3.5.3 THE PARSER DATABASE .....	43
3.6 EXPLANATION OF REGISTER DESCRIPTIONS .....	45
<b>Chapter 4 ESP REGISTERS .....</b>	<b>47</b>
4.1 XMI REQUIRED REGISTERS .....	47
4.1.1 DEVICE REGISTER (XDEV) .....	47
4.1.2 XMI BUS ERROR REGISTER (XBER) .....	48
4.1.3 XMI FAILING ADDRESS REGISTER (XFADR) .....	52
4.1.4 XMI FAILING ADDRESS EXTENTION REGISTER (XFAER) .....	52
4.2 DEMFA PORT REGISTERS .....	53
4.2.1 PORT DATA REGISTER 1 (XPD1) .....	53
4.2.2 PORT DATA REGISTER 2 (XPD2) .....	54
4.2.3 PORT STATUS REGISTER (XPST) .....	54
4.2.4 POWER UP DIAGNOSTIC REGISTER (XPUD) .....	56
4.2.5 PORT CONTROL INITIALIZE REGISTER (XPCI) .....	56
4.2.6 PORT CONTROL SHUTDOWN REGISTER (XPCS) .....	56

4.2.7 TRANSMIT CONTROL REGISTER (XTFL) .....	57
4.2.8 RECEIVE CONTROL REGISTER (XRFL) .....	58
4.2.9 COMMAND CONTROL REGISTER (XCFL) .....	58
4.2.10 UNSOLICITED CONTROL REGISTER (XUFL) .....	59
4.2.11 TRANSMIT HOST HIBERNATION LOW REGISTER (THIBL) .....	59
4.2.12 TRANSMIT HOST HIBERNATION HIGH REGISTER (THIBH) .....	60
4.2.13 RECEIVE HOST HIBERNATION LOW REGISTER (RHIBL) .....	60
4.2.14 RECEIVE HOST HIBERNATION HIGH REGISTER (RHIBH) .....	60
4.2.15 EEPROM UPDATE REGISTER (EEUP) .....	61
4.2.16 INTERRUPT VECTOR REGISTER (ITVR) .....	61
4.2.17 INTERRUPT DESTINATION VECTOR REGISTER (IDVR) .....	62
<b>4.3 ESP CONTROL AND STATUS REGISTERS .....</b>	<b>62</b>
4.3.1 ESP ERROR REGISTER (ESPERR) .....	63
4.3.2 ADAPTER MANAGER INTERRUPT REGISTER (AMIR) .....	64
4.3.3 ESP MISCELLANEOUS REGISTER (ESPMR) .....	65
4.3.4 HOST TRANSMIT RING BASE LOW ADDRESS REGISTER (HTRBLAR) .....	68
4.3.5 HOST TRANSMIT RING BASE HIGH ADDRESS AND SIZE REGISTER (HTRBHARSR) .....	69
4.3.6 HOST RECEIVE RING BASE LOW ADDRESS REGISTER (HRRBLAR) .....	69
4.3.7 HOST RECEIVE RING BASE HIGH ADDRESS AND SIZE REGISTER (HRRBHARSR) .....	70
4.3.8 PMC TRANSMIT RING BASE ADDRESS REGISTER (PMCTRBAR) .....	70
4.3.9 PMC RECEIVE RING BASE ADDRESS REGISTER (PMCRBAR) .....	71
4.3.10 ADAPTER MANAGER PACKET BUFFER ADDRESS REGISTER (AMPBAR) .....	71
4.3.11 ADAPTER MANAGER PACKET BUFFER DATA 0 REGISTER (AMPBD0R) .....	73
4.3.12 ADAPTER MANAGER PACKET BUFFER DATA 1 REGISTER (AMPBD1R) .....	73
4.3.13 ADAPTER MANAGER PACKET BUFFER DATA 2 REGISTER (AMPBD2R) .....	73
4.3.14 ADAPTER MANAGER PACKET BUFFER DATA 3 REGISTER (AMPBD3R) .....	74
4.3.15 ADAPTER MANAGER XMI LOW ADDRESS REGISTER (AMXMIL) .....	74
4.3.16 ADAPTER MANAGER XMI HIGH ADDRESS REGISTER (AMXMIH) .....	75
4.3.17 ADAPTER MANAGER XMI DATA 0 REGISTER (AMXMID0R) .....	75
4.3.18 ADAPTER MANAGER XMI DATA 1 REGISTER (AMXMID1R) .....	76
4.3.19 ADAPTER MANAGER XMI DATA 2 REGISTER (AMXMID2R) .....	76
4.3.20 ADAPTER MANAGER XMI DATA 3 REGISTER (AMXMID3R) .....	77
<b>Chapter 5 AMI REGISTERS .....</b>	<b>79</b>
5.1 ESP INTERFACE STATUS REGISTER (ESPIFSR) .....	81
<b>5.2 TIMERS AND COUNTERS .....</b>	<b>82</b>
5.2.1 AMI WATCHDOG TIMER (WDTIMR) .....	82
5.2.2 AMI RTOS TIMER (RTOSTIMR) .....	82
5.2.3 COUNTER/TIMER CSR (CTCSR) .....	83
5.2.4 BUS ERROR TIMER (BET) .....	84
5.3 AMI GENERAL PURPOSE CSR (GPCSR) .....	85
5.4 AMI EVENT STROBES .....	88

5.4.1 ADAPTER MANAGER TRANSMIT DONE STROBE (AMTDS) .....	88
5.4.2 ADAPTER MANAGER RECEIVE DONE STROBE (AMRDS) .....	89
5.4.3 SMT PRIORITY STROBE (SMTPS) .....	89
5.4.4 RTOS EVENT STROBE (RTOSES) .....	90
5.4.5 WATCHDOG TIMER CLEAR STROBE (WDTCLR) .....	90
5.5 AMI MISCELANEOUS REGISTERS .....	91
5.5.1 PARITY ERROR ADDRESS REGISTER (PEAR) .....	91
5.6 INTERRUPT AND INTERRUPT VECTOR REGISTERS .....	91
5.6.1 INTERRUPT CSR 1 (INTCSR1) .....	92
5.6.2 INTERRUPT CSR 2 (INTCSR2) .....	93
5.6.3 SOURCE INTERRUPT MASK REGISTER (SIMR) .....	95
5.6.4 IPL INTERRUPT MASK REGISTER (IPLMR) .....	96
5.6.5 SOURCE INTERRUPT ENABLE REGISTER (SIER) .....	98
5.6.6 IPL INTERRUPT ENABLE REGISTER (IPLIER) .....	99
5.6.7 INTERRUPT SELF TEST REGISTER (INTST) .....	100
5.6.8 VECTOR REGISTERS (VECREGS) .....	103
<b>Chapter 6 PMC REGISTERS .....</b>	<b>111</b>
6.1 PMC REGISTER SUMMARY .....	111
6.2 PMC REGISTER DEFINITIONS .....	117
6.2.1 PMC CONTROL AND STATUS REGISTER (CSR00) .....	117
6.2.2 PMC RAM PARITY ERROR REGISTER (CSR01) .....	119
6.2.3 PMC MASK AND INTERRUPT REGISTER (CSR02) .....	120
6.2.4 PMC PROTOCOL ERROR REGISTER (CSR03) .....	121
6.2.5 PMC PMT FAILING ADDRESS REGISTER (CSR07) .....	123
6.2.6 PMC PMT FAILING DATA REGISTER (CSR0B) .....	124
6.2.7 PMC PMT CONTROL AND STATUS REGISTER (CSR0F) .....	124
<b>Chapter 7 PARSER REGISTERS AND DATABASE .....</b>	<b>127</b>
7.1 CSR REGISTERS .....	127
7.1.1 CONTROL REGISTER 0 (CR0) .....	127
7.1.2 CONTROL REGISTER 1 (CR1) .....	129
7.1.3 STATUS REGISTERS .....	130
7.1.3.1 INTERRUPT CAUSE REGISTER (ICR) .....	130
7.1.3.2 INTERRUPT MASK REGISTER (IMR) .....	131
7.1.4 DATABASE UPDATE REGISTER(S) .....	131
7.2 PARSER DATABASE .....	135
7.2.1 FRAME CONTROL .....	135
7.2.2 DESTINATION ADDRESSES .....	136
7.2.3 FILTERING DATA FOR FC AND DA ONLY FILTERING .....	136
7.2.4 DESTINATION SERVICE ACCESS POINT (DSAPs) .....	137
7.2.5 PROTOCOL IDENTIFICATIONS (PIDS) .....	138
7.2.6 FILTERING DATA FOR FC,DA AND LLC FILTERING .....	138

7.2.7 FILTERING DATA FOR A PARTICULAR USER FOR FC,DA AND LLC FILTERING .....	139
7.2.8 FILTERING DATA FOR PROMISCUOUS USER(S) .....	140
<b>Chapter 8 FDDI REGISTERS .....</b>	<b>143</b>
<b>Chapter 9 DEMFA MISCELLANEOUS FEATURES .....</b>	<b>145</b>
9.1 DEMFA MODULE DATA INTEGRITY .....	145
9.2 FDDI MAINTENANCE SERVICES .....	145
9.3 INTERLOCKED OPERATION .....	146
9.4 SELF TEST PROCEDURE .....	147
9.5 STATUS LEDS .....	147
<b>Chapter 10 DEMFA MODULE SPECIFICATION .....</b>	<b>149</b>
10.1 DEMFA MODULE SPECIFICATION .....	149
10.1.1 OVERVIEW .....	149
10.1.2 POWER DISSIPATION .....	149
10.1.2.1 DEMFA XMI MODULE .....	149
10.1.2.2 DEMFA ACTIVE BULKHEAD MODULE .....	149
10.1.3 DEMFA CLOCK .....	149
10.1.4 MECHANICAL DIMENSIONS .....	150
10.1.4.1 DEMFA XMI MODULE .....	150
10.1.4.2 DEMFA ACTIVE BULKHEAD .....	150
10.1.5 OPERATING ENVIRONMENT .....	150
10.1.6 NON-OPERATING ENVIRONMENT .....	151
10.1.7 MEANTIME BETWEEN FAILURES .....	151
<b>Appendix A DEMFA MODULE I/O PINNING .....</b>	<b>153</b>
A.1 XMI INTERFACE SIGNALS .....	153
A.2 DEMFA ACTIVE BULKHEAD INTERFACE SIGNALS .....	156
<b>Appendix B GATE ARRAY PINNING .....</b>	<b>159</b>
B.1 ESP GATE ARRAY PINNING .....	159
B.2 PMC GATE ARRAY PINNING .....	161
B.3 AMI GATE ARRAY PINNING .....	163
B.4 PAR GATE ARRAY PINNING .....	165
<b>Appendix C ACTIVE BULKHEAD .....</b>	<b>167</b>
C.1 INTRODUCTION .....	168
C.2 RELATED DOCUMENTS .....	168

C.3	PRODUCT DESCRIPTION .....	168
C.3.1	PARTSLIST .....	169
C.3.2	FEATURES .....	170
C.3.3	COMPONENT LOCATION .....	170
C.3.4	DEMFA BLOCK DIAGRAM .....	171
C.3.5	SIGNAL DEFINITIONS .....	171
C.3.6	TESTABILITY FEATURES .....	172
C.3.7	OPERATION .....	172
C.4	RECOMMENDED OPERATING CONDITIONS .....	172
C.4.1	ENVIRONMENTAL CONDITIONS .....	172
C.4.2	DC CHARACTERISTICS .....	172
C.4.3	TERMINATION REQUIREMENTS .....	172
C.5	DEVICE PINOUTS .....	173
C.5.1	CONNECTOR PINOUT .....	173
C.5.2	CDCR PINOUT .....	174
C.5.3	CDCT PINOUT .....	175
C.5.4	FIBER OPTIC RCVR PINOUT .....	176
C.5.5	FIBER OPTIC XMITTER PINOUT .....	176
C.6	T2027 MODULE REQUIREMENTS .....	177
C.7	ISSUES/MISCELLANEOUS .....	178
<b>Appendix D PARSER ALGORITHM .....</b>		<b>179</b>
 <b>FIGURES</b>		
1	FDDI CONFIGURATION .....	6
2	XMI - FDDI ADAPTER BLOCK DIAGRAM .....	10
3	DEMFA PERFORMANCE .....	14
4	DEMFA RESPONSE TO XMI COMMANDS .....	16
5	DEMFA INITIATED XMI COMMANDS .....	17
6	ESP FUNCTIONAL BLOCK DIAGRAM .....	21
7	68020 SUBSYSTEM - BLOCK DIAGRAM .....	23
8	ESP BUS READ .....	24
9	ESP BUS WRITE .....	25
10	AMI BLOCK DIAGRAM .....	26
11	PACKET MEMORY CONTROLLER - FUNCTIONAL BLOCK DIAGRAM .....	31
12	MEMORY ALLOCATION .....	33
13	FDDI CHIP SET .....	36
14	PARSER SUBSYSTEM IN DEMFA .....	40
15	FORWARDING VECTOR .....	41
16	PARSER SUBSYSTEM BLOCK DIAGRAM .....	44
17	MEMORY MAP .....	80
18	PARSER DATABASE MEMORY ALLOCATION .....	142
19	THE DEMFA DATA INTEGRITY SCHEME .....	146
20	EXTERNAL DIAGNOSTIC CSR REGISTER .....	148



21	ESP GATE ARRAY PIN DIAGRAM .....	160
22	PMC GATE ARRAY PIN DIAGRAM .....	162
23	AMI GATE ARRAY PIN DIAGRAM .....	164
24	PARSER GATE ARRAY PIN DIAGRAM .....	166
25	XMI to FDDI Adaptor Module Component Location .....	170
26	Block Diagram of the XMI Node, Active Bulkhead, and FDDI .....	171
27	Block Diagram of the CDCR Connections .....	174
28	Block Diagram of the CDCT Connections .....	175
29	THE PARSING ALGORITHM - FC Filtering .....	180
30	THE PARSING ALGORITHM (CONTD.) - DA Filtering .....	181
31	THE PARSING ALGORITHM (CONTD.) - FCDA Filtering .....	182
32	THE PARSING ALGORITHM (CONTD.) - LLC Filtering .....	183
33	THE PARSING ALGORITHM (CONTD.) - FCDALLC Filtering .....	184
34	THE PARSING ALGORITHM (CONTD.) - Non SNAP SAP Filtering .....	185
35	THE PARSING ALGORITHM (CONTD.) - Class1 XID/TEST Filtering .....	186
36	THE PARSING ALGORITHM (CONTD.) - NULL DSAP Filtering .....	187
37	THE PARSING ALGORITHM (CONTD.) - SNAP DSAP Filtering .....	188
38	THE PARSING ALGORITHM (CONTD.) - Promiscuous User Filtering .....	189
39	THE PARSING ALGORITHM (CONTD.) - User Validity Filtering .....	190
40	THE PARSING ALGORITHM (CONTD.) - Buffer Descriptor Filtering .....	191

## TABLES

1	DEMFA PERFORMANCE .....	12
2	PBI INTERFACE COMMAND/ADDRESS BIT DEFINITIONS FOR DATA MOVER .....	18
3	PBI INTERFACE COMMAND/ADDRESS BIT DEFINITIONS FOR ADAPTER MANAGER .....	18
4	INTERRUPT STRUCTURE .....	28
5	COLOR AND RING ASSIGNMENTS .....	32
6	PMC CSR REGISTER ALLOCATION .....	111
7	DEMFA MODULE STATUS LEDS .....	147
8	ELECTRICAL CHARACTERISTICS .....	149
9	ELECTRICAL CHARACTERISTICS .....	149
10	XMI MODULE DIMENSIONS .....	150
11	OPERATING ENVIRONMENTAL REQUIREMENTS .....	150
12	NON-OPERATING ENVIRONMENTAL REQUIREMENTS .....	151
13	DEMFA MTBF (HOURS) .....	151

# CHAPTER 1

## DEMFA INTRODUCTION

This document contains the functional specification of the DEMFA, an XMI to FDDI adapter, being a description of the inherent functionality contained within the boundary of the XMI bus and the FDDI physical medium (optical fiber). The reader of this document is presumed to have some knowledge of the DEMFA Port Architecture. No attempt is made here to duplicate Port Architecture except as it directly relates to the hardware, eg, the Host visible register set. However, a high level understanding of the DEMFA *functionality* may be gotten from Chapter 2, the executive summary, where no prior knowledge is assumed.

This document is intended as volume 1 of a 2 volume set, the second volume being the DEMFA Firmware Functional Specification. The Firmware Specification includes descriptions of (among others) Adapter Initialization, Adapter States, Frame and Command processing and updating firmware. These descriptions will not be duplicated in this volume.

Included for completeness in Appendix C of this volume is the Functional Specification of the active bulkhead.

### 1.1 INTENDED AUDIENCE

This document is intended for the following audiences:

- Designers of the DEMFA Functional firmware
- Designers of the DEMFA Self\_Test firmware
- Designers of other DEMFA diagnostics
- Manufacturing
- CSSE
- 'Casual' Readers

### 1.2 GOALS AND NON-GOALS

It is the goal of this document to satisfy the needs of the above audiences. As such this document will contain:

- An executive summary for those readers only interested in a product overview
- A high level summary of the Port Functionality
- A high level functional description to give the reader an understanding of the major imbedded functions and their relationships.
- An exhaustive description of the Host visible registers
- An exhaustive description of the internal microprocessor visible registers
- Descriptions of the gate arrays sufficient for the intended audiences

- A summary of the Maintenance Features
- A summary of how data integrity is maintained and how errors are handled.
- The DEMFA module specification
- The Active Bulkhead specification
- Appendices containing other miscellaneous information
- Some information concerning performance

It is *NOT* the goal of this document to:

- Provide details of the DEMFA Port Specification
- Provide descriptions of FDDI Registers
- Provide design details
- Provide details of any other 'non-functionality', eg: manufacturing, schedule, module layout, etc, etc...

### 1.3 ASSOCIATED DOCUMENTS

- DIGITAL: DEMFA Port Specification, Revision 2.0
- DIGITAL: DEMNA Port Specification, Revision 4.0, Jan-1989.
- DIGITAL: XI Data Link Architecture Specification, Version X0.1.2, Apr-1987 (Will eventually include FDDI)
- DIGITAL: Calypso Memory Interconnect (XMI), Revision 1.4, 28-Nov-1988.
- DIGITAL Network Architecture CSMA/CD Data Link Functional Specification, T2.0.0, 31-Mar-1989.
- DIGITAL NI Node Product Architecture Specification, T2.0.1, 11-Nov-1988. (Will eventually include FDDI)
- DIGITAL Network Architecture Maintenance Operations Functional Specification, T4.0.0, 28-Jan-1988.
- ANSI FDDI - PMD Standard, Rev 7.3, May 20, 1988.
- ANSI FDDI - PHY Standard, Rev 15, September 1, 1987.
- ANSI FDDI - MAC Standard, Rev DRAFT 7/1/87.
- ANSI FDDI - SMT Standard, Rev 3.0, August 1, 1987.
- IEEE STD 802.2 Logical Link Control, 1985.
- DIGITAL: Clock and Data Conversion Transmitter Chip Specification, Rev 1.1, OCT 22, 1987.
- DIGITAL: Clock and Data Conversion Receiver Chip Specification, Rev 1.2, FEB 8, 1988.
- DIGITAL: Elasticity Buffer and Link Management Chip Specification, (in progress)
- DIGITAL: Media Access Control Chip Specification, Rev 1.1, NOV. 6, 1987.
- DIGITAL: Ring Memory Controller Chip Specification, Rev 2.0, Nov 4, 1988.
- DIGITAL: ESP Gate Array Design Specification, (in progress)

## 2 DEMFA INTRODUCTION

- DIGITAL: PMC Gate Array Design Specification, (in progress)
- DIGITAL: AMI Gate Array Design Specification, (in progress)
- DIGITAL: PAR Gate Array Design Specification, (in progress)
- DIGITAL: DEMFA Firmware Functional Specification, Version X1.2
- DIGITAL: DEMFA Firmware Design and Test Specification (in progress)
- DIGITAL: DEMFA Performance Simulation Studies Document (in progress)

## 1.4 TERMINOLOGY

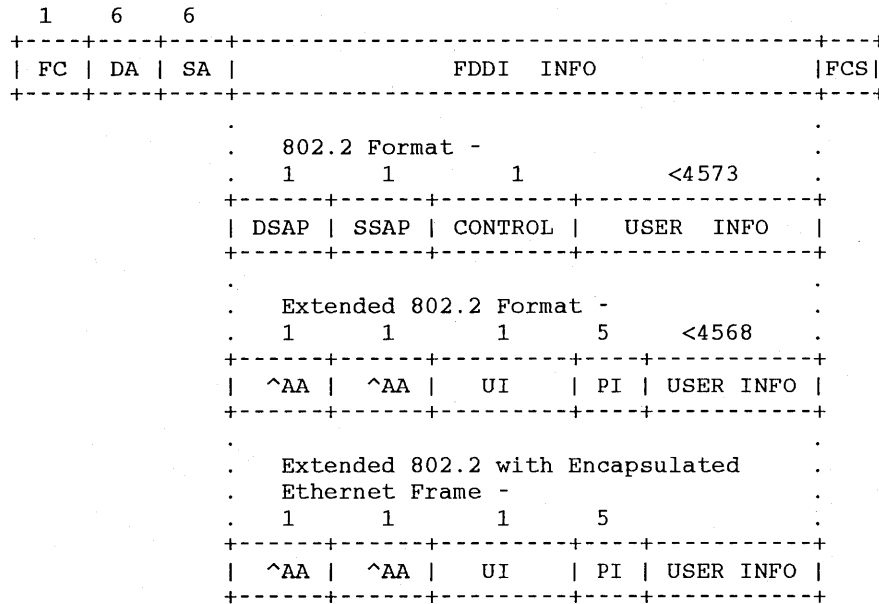
Some terms used in this document are defined here.

- **FRAME and PACKET.** Within this document these terms are used somewhat interchangeably. This is not strictly correct. In general, references to 'Packet' are more correctly 'Frame'.
- **FDDI.** Fiber Distributed Data Interface. The ANSI Specifications.
- **802.** 802 refers to the class of local area networks defined in IEEE 802 specifications.
- **802 SAP.** An FDDI frame using 802.2 connectionless data link layer would contain, as the first information bytes, destination and source identifiers (each called a SAP, a Service Access Point), one byte control field followed by the user data. The source SAP(SSAP) identifies the user sending the frame. The destination SAP(DSAP) identifies the user to receive the frame.
- **802 SNAP.** In the case where the destination and source SAPS consists of alternating ones and zeros, AA(HEX), and the control is UI (Unnumbered Information) this is called the SNAP SAP(Subnet Access Protocol SAP). This implies an additional five byte protocol identifier(PI) field will follow the control field. The five byte PI field identifies the particular user.
- **Adapter.** The adapter is the hardware and firmware entity which comprises the DEMFA module. It provides an interface between Host system Port Driver/Memory resident data structures and the FDDI ring via the XMI bus.
- **Port.** The port is the entity on the adapter which embodies the port architecture. It performs actions requested of it by the port driver.
- **Port driver.** The port driver is the software running on the host processor which directs the operation of the port. It processes transmit and receive request from users of the operating system(or the operating itself).
- **Host Ring.** An Host ring is the type of data structure used to communicate buffer information between the port driver and the port. After initialization, the port driver and the port circle the ring processing commands, responses, transmit requests and received frames.
- **Host Receive Ring.** A ring located in host memory. After port initialization it has fixed size and location. It is used to communicate port driver receive buffers to the port for the use as repositories of frames received from the FDDI network and intended for host delivery.

- Host Transmit Ring. A ring located in host memory. After port initialization it has a fixed size and location. It is used to communicate port driver transmit buffers to the port. Those buffers are intended for data which the adapter transmits in a frame on the FDDI ring.
- Host Command Ring. A ring located in host memory. After port initialization it has a fixed size and location. It is used to communicate port driver command buffers to the port. Those buffers are specifically commands intended for the adapter manager, for example get statistics, use these parameters, start this user, etc.
- Host Unsolicited Ring. A ring located in host memory. After port initialization it has a fixed size and location. It is used to communicate unsolicited (ie, by the device driver) information from the Adapter to the Device Driver. eg, congestion information.

### 1.5 FDDI/802.2 Frame Format

Following is a representation of the FDDI/802.2 frame format where FC = Frame Control, DA = Destination Address, SA = Source Address and FCS = Frame Check Sequence. For a full definition of these terms refer to the ANSI MAC Specification.



PI format for Encapsulated Ethernet contains the Ethernet Type field in the low order two bytes.

## CHAPTER 2

### DEMFA EXECUTIVE SUMMARY

The DEMFA adapter is a high speed controller that connects the XMI bus to the Fiber Distributed Data Interface (FDDI). The DEMFA adapter and its associated device driver implement the FDDI NODE architecture, the FDDI Data Link architecture the DEMFA Port Specification.

The DEMFA is a single attachment station.

The DEMFA expects all host memory addresses passes to it to be *physical addresses*, ie, the adapter does not understand host virtual address schemes.

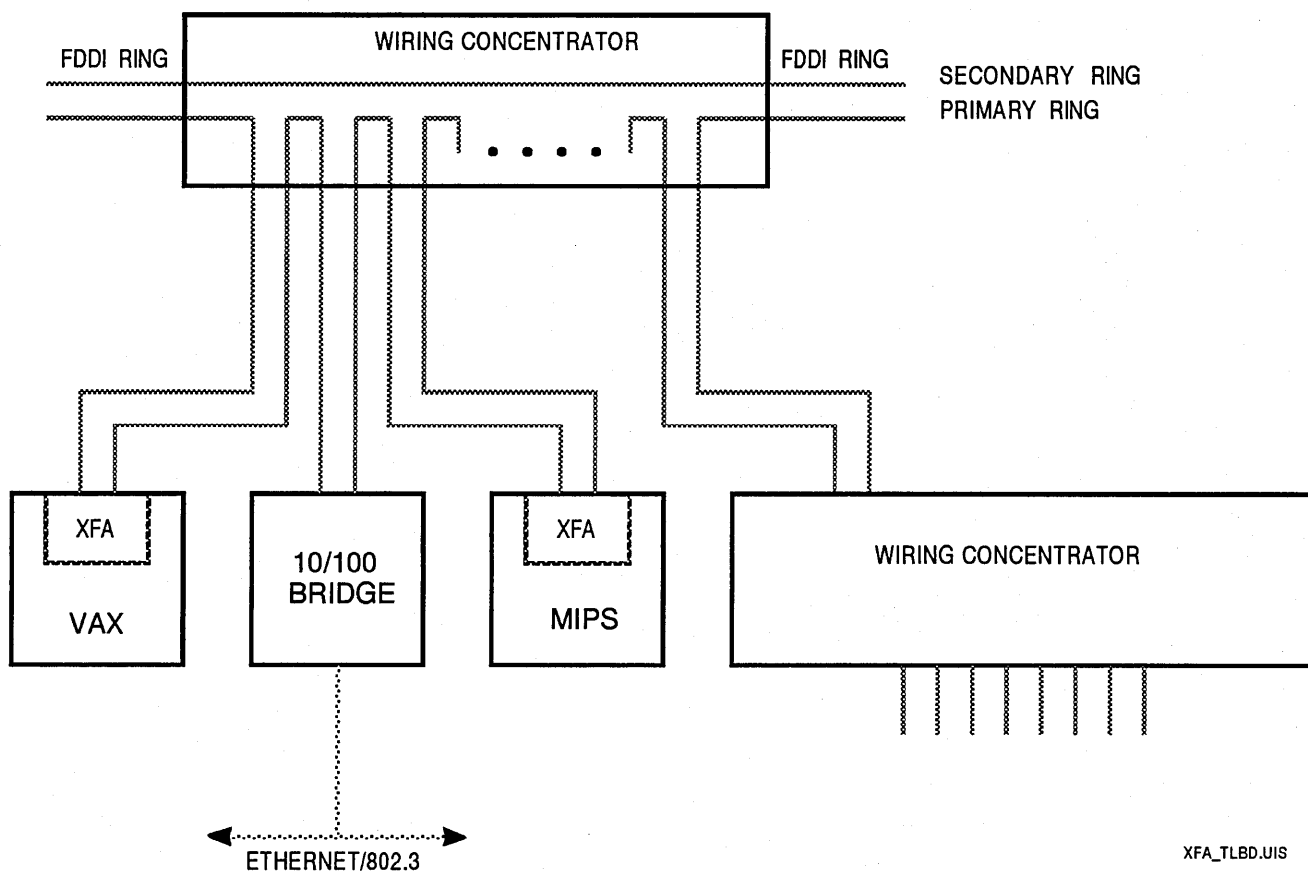
The DEMFA is compatible with both XMI-1 and XMI-2.

The technology chosen for two out of four gate arrays (ESP and PMC) is 1.0u Sea of Gates (SOG) and for other two gate arrays (AMI and PARSER) is 1.5u Sea of Gates (SOG). The RMC and Address CAM chips are 1.5u Hudson Semi-custom. The MAC and ELM chips are also of 1.5u Sea of Gates (SOG).

#### 2.1 DEMFA CONTEXT

The DEMFA is being developed by VAX products and options group in Littleton. The first wave of Digital FDDI products will include an DEMFA for connecting XMI based CPUs and servers to FDDI, a bridge for connecting 10Mb/s Ethernet/IEEE802.3 LANs to FDDI and a wiring concentrator which will provide the physical connection for the DEMFA and bridge to the FDDI ring. A typical connection among DEMFA, bridge and wire concentrator is shown in Figure 1.

Figure 1: FDDI CONFIGURATION



## 2.2 DEMFA OVERVIEW

The DEMFA can be logically divided into 4 parts:

1. Host interface
2. Adapter Manager subsystem
3. Packet buffer memory subsystem, including the Ring Entry Mover (REM).
4. FDDI chipset and Parser

## 2.2.1 HOST INTERFACE

The Host interface to the XMI consists of the standard XMI corner and one Sea-of-Gates gate array (the ESP). The XMI corner contains the XMI specific XCLOCK and XLATCH chips that interface to the XMI backplane. The ESP Gate Array includes a single XMI commander and Responder, two Hexaword buffers for high speed data movement and hardware that understands how to transmit and receive packets without microprocessor intervention. The DEMFA implements Hex Writes and the More bit protocol. For more details see Chapter 3.

## 2.2.2 THE ADAPTER MANAGER SUBSYSTEM

A 68020 microprocessor together with memory and a support logic gate array (the AMI gate array) provide what we call the adapter manager functions. These functions include:

- Initialization and event/error handling
- Processing and responding to device driver commands
- Processing and responding to FDDI packets not generally deliverable to the host, ie, Station Management and MOP packets.
- Local management of the FDDI MAC, PHY, and PMD sublayer entities within the DEMFA.
- Collection and Maintenance of statistics

The Adapter Manager subsystem consists of a 68020 microprocessor with 256K bytes of EEPROM for functional and self-test microcode, 256K bytes of SRAM for operational code and scratch pad, a DPA ROM (Default Physical Address ROM), and the AMI gate array. The AMI gate array contains timers, interface logic to the FDDI chipset, and interface logic to the host and packet buffer memory (via the ESP gate array).

## 2.2.3 THE PACKET BUFFER MEMORY SUBSYSTEM

The Packet buffer memory subsystem is logically a four ported memory and physically a three ported memory. The three physical ports are the ESP, the RMC (part of the FDDI chipset) and the REM (Ring Entry Mover - a function internal to the PMC gate array). The three physical interfaces are logically identical to the RMC. Two logical ports, one for the host interface and one for the adapter manager are multiplexed across the ESP physical interface.

Packets exist in virtual space, being indirectly addressed through a page table lookup, pages being 512 bytes. Packets are organized in 6 groups, being a transmit group and a receive group for each of the RMC, host interface and adapter manager. Each group is controlled by a ring structure, being a circular list of control elements, each control element containing page indirection, packet specific information (length, destination, status, etc) and ownership.

The Ring Entry Mover is a programmable functional element within the PMC gate array whose function is to move packets between the various interfaces. It accomplishes this by manipulating the control elements. Packet data is never copied, ie the source interface writes the packet and the destination interface reads the packet, only control information 'moves'. For example, the RMC has moved a packet into the packet memory. Once the RMC relinquishes ownership of this page or set of pages the REM will look at some fields in the control element to determine where this page should go. The page could be discarded, sent to the adapter manager or to the host. Once the REM has moved the control information it will relinquish ownership of this page. This action makes the packet available to the



adapter manager (or the host interface) which will take the appropriate action. A detailed description of how the ring structures work can be found in the RMC chip specification.

The packet buffer memory subsystem consists of 1M bytes of DRAM for packet buffer memory, 64K bytes of SRAM for the control elements, being the Page Table Entries and Buffer Descriptors, and one Sea-Of-Gates gate array, (the PMC)

#### 2.2.4 THE FDDI CHIPSET AND THE PARSER

The FDDI chipset provides the actual interface to the FDDI. The chipset consists of the Ring Memory Controller (RMC), the Media Access Controller (MAC), the Elasticity buffer and Link Manager (ELM), the transmit CDC (TX CDC), the Receive CDC (RCV CDC), and the Fiber Optic transmitter and receiver chips (FOX). These chips have been developed by NaC and will also be used by other products, such as the wiring concentrator and the bridge. Following is a list of the major features of these chips.

The major features of the RMC (Ring Memory Controller) gate array are:

- Independent transmit and receive channels
- Fragment filter to discard small "non-packets"
- 4 or 8 longword burst mode to packet buffer memory
- Circular (ring) buffer support

The major features of the MAC (Media Access Control) gate array are:

- Respond to the frame Control byte for received frames
- Destination Address filtering
- Perform CRC checking for receive packets
- Save frame status indicators
- Keep a count of the good and bad frames received
- Assemble packets for transmission
- Generate CRC for transmit packets
- Can internally generate and send Claim, Beacon, and Void frames.

The major features of the ELM (Elasticity Buffer and Link Management) gate array are:

- Elasticity Buffer (for synchronizing recovered and local clocks)
- 4B/5B encoding/decoding
- Line State Detection
- Repeat filter for MAC-less connection
- Physical Connection Management state machines

The TX CDC converts the parallel data stream into serial data stream before it sends the data to the Fiber Optic transmitter.

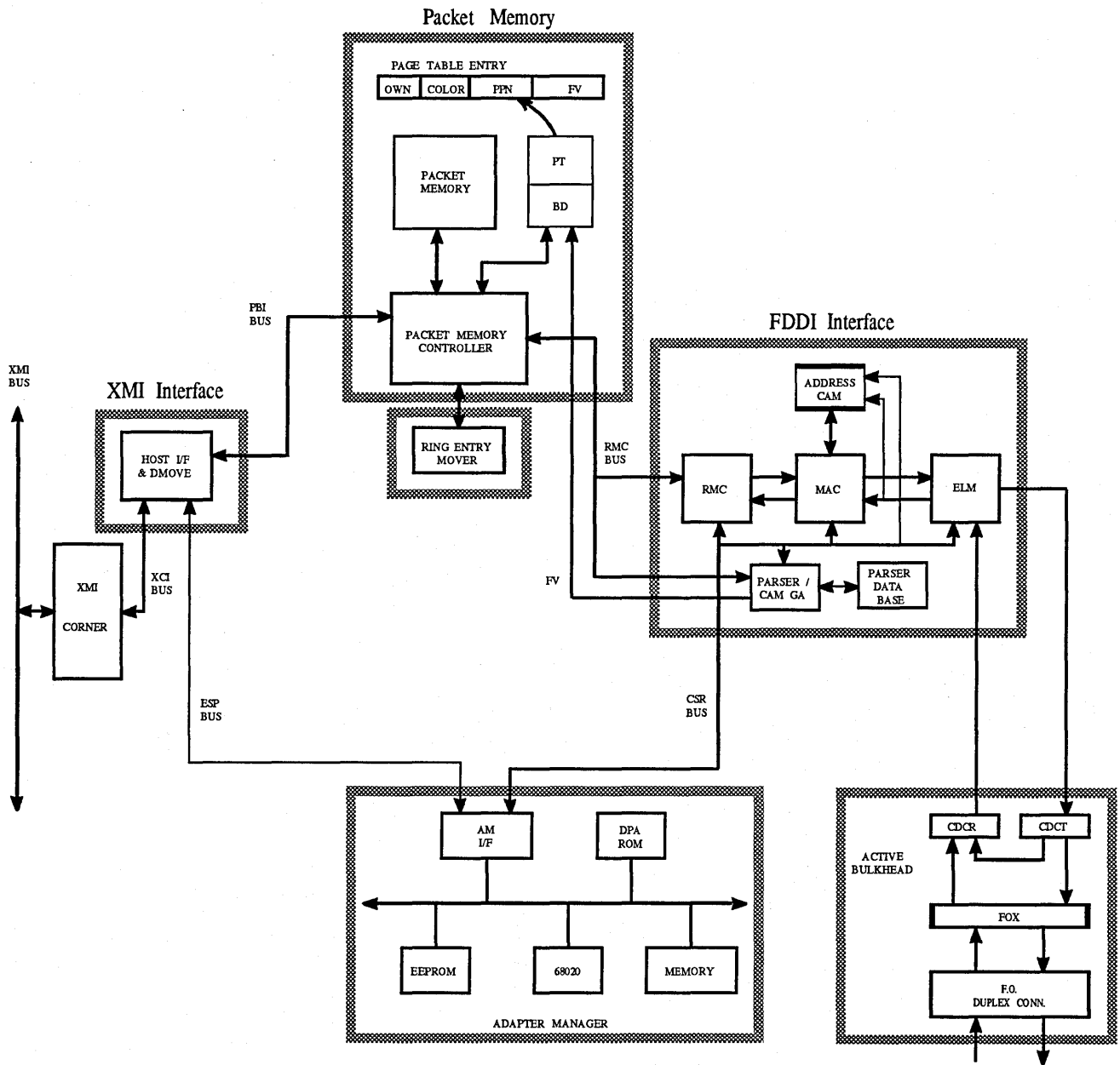
The RCV CDC converts the serial data stream it receives from the Fiber Optic receiver into a parallel data stream destined for the ELM chip.

The FOX is the combination of the Fiber Optic transmitter and receiver.

The last three components, TX CDC, RCV CDC, and FOX, reside on what is called an active bulkhead module, which is attached to an I/O bulkhead plate. The I/O bulkhead plate is mounted on the I/O panel of the CPU cabinet. The fiber optic connector from the FDDI ring plugs into this I/O bulkhead plate.

The Parser gate array snoops on the RMC bus and does packet parsing based on information in a local (to the parser) data base during the receive of a packet from the RMC. It generates a Forwarding Vector which is written into the control element, Page Table Entry, associated with each page of a packet.

Figure 2: XMI - FDDI ADAPTER BLOCK DIAGRAM



XFA\_BLOCK\_DIAGRAM\_3.UIS

## 2.3 PACKET RECEPTION

Following is a brief description of how a packet is received from the FDDI physical medium and delivered to the user portal (the data link interface). Please refer to Figure 2.

Photons received from the optical fiber are converted to electrical signals by the FOX.

The electrical signals are passed to the CDCR for conversion to signals suitable for a CMOS gate array, viz the ELM. The CDCR also recovers the imbedded clock.

The ELM synchronizes the data with the local clock. The ELM chip decodes the encoded data stream into data or control symbols. The decoded data is passed to the MAC chip who analyses the data stream.

In the case of packets for this endnode, the MAC recognises it as such, the MAC passes the packet to the RMC for delivery to the Packet Memory. The MAC also all the time checks the packet for good CRC. The MAC generates byte parity.

The RMC will deliver the packet to the Packet Memory in a series of data bursts and followed by a BD (Buffer Descriptor which contains the packet status and length) write then a 'clear ownership' cycle (ownership of the packet buffer is passed to the next stage of processing) and finally status is passed to the Ring Entry Mover that a packet has been completely delivered.

The Parser, who 'snoops' on the RMC bus, parses the packet and, depending on his local data base, will generate a Forwarding Vector indicating what is to be done with the packet, ie, discard, deliver to host, deliver to adapter manager, for use by the ring entry mover in the PMC gate array.

The Packet Memory Controller provides the necessary interfaces to the packet memory and the page and buffer descriptor tables. The PMC also arbitrates between the interfaces. The PMC contains the Ring Entry Mover (REM). The Ring Entry Mover is made aware of the delivery of a packet from the FDDI by the RMC. Upon receipt of this indication, the REM analyses the forwarding vector and 'moves' the packet to the appropriate place for delivery (or discards the packet). The REM also manages the buffer resources and gathers statistics on discarded packets.

In this case (packet from FDDI to Host User Interface), the REM alerts the XMI interface (the ESP) that a packet has been queued for delivery. Assuming the ESP has already got host buffer(s) for delivery of the packet, the ESP will set up his dual hexaword buffer data mover engine to deliver the packet. Upon completion of the data delivery, the ESP will deliver appropriate status and pass ownership of the packet to the device driver. If it is necessary to 'wake up' the device driver to process the packet, a host interrupt will be sent. Great care is taken not to interrupt the host unnecessarily. Upon receipt of the interrupt, the device driver will process the packet and pass it to the appropriate user via a data link portal.

## 2.4 PACKET TRANSMISSION

Following is an example of a packet transmission from the User Data link interface to the FDDI physical medium. Users pass packets for transmission to the device driver via a data link portal. The device driver in turn queues the packet for transmission and, if necessary, alerts the ESP gate array of the fact. The ESP sets up the dual hexaword buffer data mover (shared with receive path) to move the data between host memory and the packet buffer.

Upon completion of the move, the ESP writes appropriate packet status to both the Host and the Packet Memory. The ESP alerts the REM that a packet has been queued in the packet buffer.

The REM understands that packets from the host are only intended (except in maintenance mode) for transmission over the FDDI. The REM therefore make the necessary 'move' by manipulating the corresponding control elements. Once the REM has completed the 'move', it informs the RMC that a packet has been queued for transmission over the FDDI.

The RMC reads the first part of the packet and passes the first four bytes of the packet to the MAC. This is the packet header and frame control information (supplied by the device driver) and provides the MAC with the necessary information to capture a token and some other packet specific information (refer to the RMC and MAC Specification for details). Once the MAC informs the RMC that the token has been captured, the RMC fetches the rest of the packet data. Upon completion of the packet transmission, the RMC, based on internal and MAC derived information, passes packet status to the Packet Memory for storage in the appropriate buffer Descriptor(s).

The MAC, as previously alluded, is responsible for capturing an appropriate token and, upon capture, allow transmission of any packet(s) queued up to a predetermined limit. The MAC (on a per packet basis) may optionally generate CRC which is appended to the packet. The MAC passes the packet to the ELM.

The ELM encodes each nibble (symbol in FDDI parlance) according to 4B/5B encoding and passes the symbol-wide data stream to the CDCT.

The CDCT serializes the data stream and converts the signal levels to that appropriate to the FOX.

The FOX converts the electrical signals to analogous optical signals which are transmitted into the FDDI physical medium.

## 2.5 PERFORMANCE

Performance is summarized by the following figure, Figure 3. The figure shows three performance metrics.

Firstly, the adapter is capable of sustaining minimum packet traffic with the FDDI (up to the limit of the packet buffer).

Secondly, the following table shows the estimated performance across the XMI bus.

**Table 1: DEMFA PERFORMANCE**

SYSTEM	Receive Packet Size (bytes)	Receive Bandwidth (MBytes)	Transmit Packet Size (bytes)	Transmit Bandwidth (MBytes)
Calypto <sup>1</sup>	150	11.87	150	11.87
Aquarius <sup>2</sup>	150	11.87	1500	12.43

<sup>1</sup>Minimal Contention on the XMI

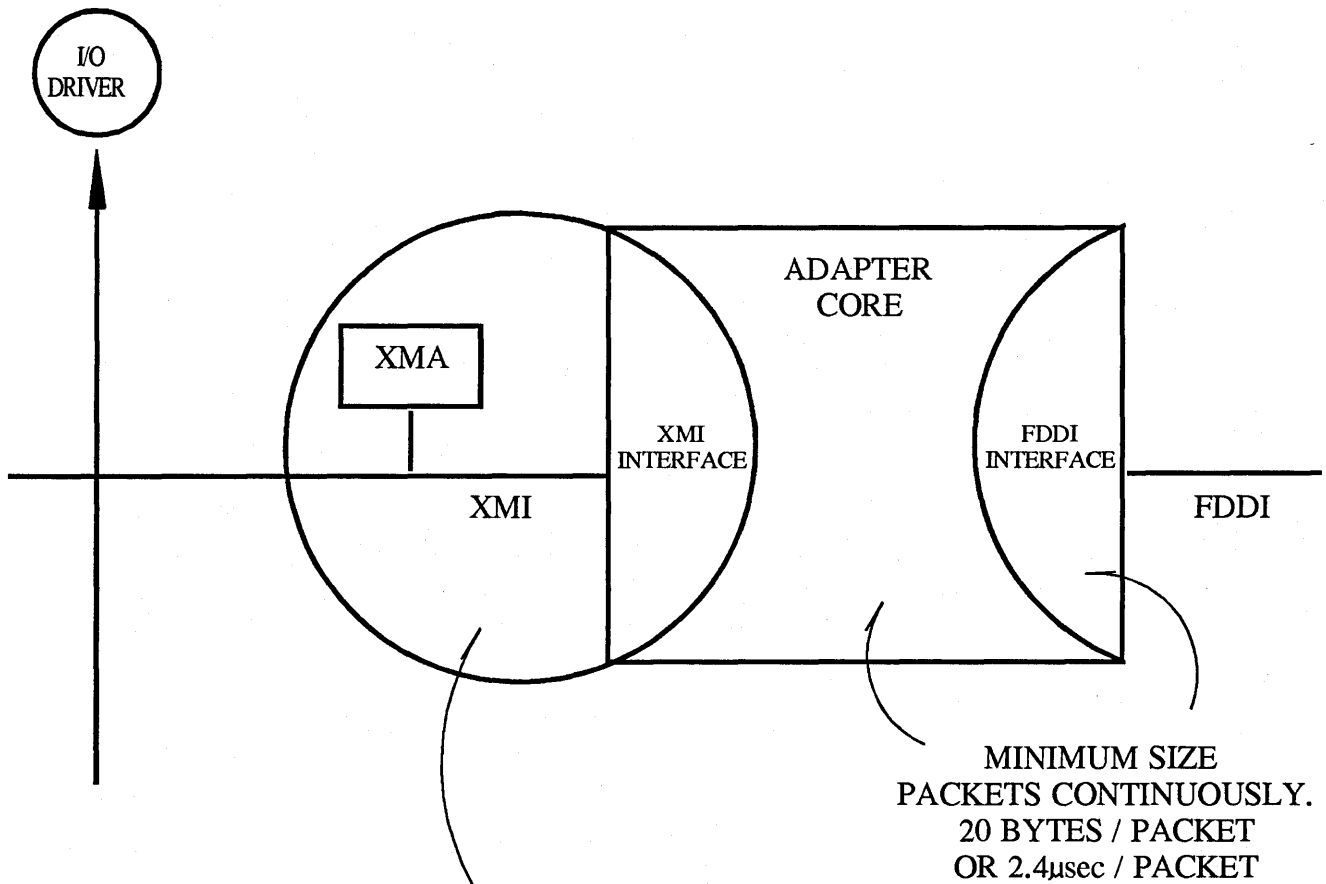
<sup>2</sup>Minimal XMI and J\_Bus contention. Aquarius number for transmit packet size is large because number of cycles for hexaword read is relatively large

The performance given in Table 1 is calculated making some assumptions. For full details refer to the DEMFA Performance Simulation Studies document.

Thirdly, since the I/O driver can process packets at the rate of 1.5 to 2 msec per packet per VUP, this implies that packets may be transferred across the user (data link) interface at the rate of 500 packets/sec/vup approximately. Not shown in the diagram is latency. Latency is less than 20 microseconds assuming system lightly loaded, no queueing delays, first bit to last bit for a 22 byte packet (that is, first bit received from the optical fiber to last bit delivered to host memory).

Figure 3: DEMFA PERFORMANCE

SUBSYSTEM LEVEL PERFORMANCE



1.5 to 2 msec./packet  
for 1 vup machine.

500 packets/sec/vup.

150 BYTE PACKETS CONTINUOUSLY.  
(FUNCTION OF HOST MEMORY)

MINIMUM SIZE  
PACKETS CONTINUOUSLY.  
20 BYTES / PACKET  
OR 2.4μsec / PACKET

# CHAPTER 3

## DEMFA FUNCTIONAL DESCRIPTION

The DEMFA module is logically divided into four parts:

1. Host Interface
2. Adapter Manager Subsystem
3. Packet Buffer Subsystem, including the Ring Entry Mover
4. FDDI chipset (excluding the active bulkhead) and Parser

### NOTE

**Functional Specification of the active bulkhead is included in Appendix C**

### 3.1 HOST INTERFACE

#### 3.1.1 THE XMI CORNER

A detailed description of the XMI corner may be found in the XMI Specification. In summary, the corner provides buffering between the XMI and the adapter, and provides the adapter with a six phase clock, synchronous with the rest of the XMI subsystem.

#### 3.1.2 DEMFA SUPPORTED XMI TRANSACTIONS

Following are summaries of the DEMFA responses to XMI commands and DEMFA initiated XMI Commands.



Figure 4: DEMFA RESPONSE TO XMI COMMANDS

XMI D	XMI CMD	XMI COMMAND DESCRIPTION	DEMFA'S XMI COMMAND CONFIRMATION	FUNCTION EXECUTED BY DEMFA	N O T E
X X	0000	RESERVED	NACK	NONE	1
H H	0001	READ	NACK	NONE	
H L	0001		ACK (+ GRD later)	READ LW	3
L H	0001		NACK	NONE	
L L	0001		NACK	NONE	
H H	0010	INTERLOCK READ	NACK	NONE	
H L	0010		ACK (+ GRD later)	READ LW	2
L H	0010		NACK	NONE	
L L	0010		NACK	NONE	
X X	0011	RESERVED	NACK	NONE	1
X X	0100	RESERVED	NACK	NONE	1
X X	0101	RESERVED	NACK	NONE	1
H H	0110	UNLOCK WRITE MASK	NACK	NONE	1
H L	0110		ACK(/NACK if busy)	WRITE LW	3
L H	0110		NACK	NONE	
L L	0110		NACK	NONE	
H H	0111	WRITE MASK	NACK	NONE	
H L	0111		ACK(/NACK if busy)	WRITE LW	3
L H	0111		NACK	NONE	
L L	0111		NACK	NONE	
H H	1000	INTERRUPT	NACK	NONE	
H H	1001	IDENT	ACK	IDENTIFY	
X X	1010	RESERVED	NACK	NONE	1
X X	1011	RESERVED	NACK	NONE	1
X X	1100	RESERVED	NACK	NONE	1
X X	1101	RESERVED	NACK	NONE	1
X X	1110	RESERVED	NACK	NONE	1
H H	1111	IMP. VEC. INTR.	NACK	NONE	

NOTE: X = don't care

1. The DEMFA's XMI interface will NACK all XMI RESERVED commands.
2. The DEMFA will ACK interlocked transactions but no lock bit is implemented. Interlock reads are treated like non-interlocked reads. Unlock write masks are treated like write masks.
3. All unassigned addresses within the XMI Node Address space will be NACKed.

**Figure 5: DEMFA INITIATED XMI COMMANDS**

DEMFA COMMAND DESCRIPTION	XCI D	XMI CMD	XMI COMMAND DESCRIPTION
	<31: 30>	CODES <63:60>	
READ OCTAWORD	1 1	0001	READ
READ HEXAWORD	0 0	0001	READ
WRITE MASK QUADWORD	1 0	0111	WRITE MASK
WRITE MASK OCTAWORD	1 1	0111	WRITE MASK
WRITE MASK HEXAWORD	0 0	0111	WRITE MASK
INTERRUPT	0 0	1000	INTR

NOTE: A Hexaword write cannot be masked.

The following transactions are supported by the XMI interface with two possible priority configurations.

PRIORITY CONFIG. 1 (EM_Prio=0)	PRIORITY CONFIG. 2 (EM_Prio=1)
HOST READ/WRITE (LW)	HOST READ/WRITE (LW)
IDENT RESPONSE (LW)	IDENT RESPONSE (LW)
INTERRUPTS	INTERRUPTS
WRITE-BACK RECEIVE BUFFER DESCRIPTOR PROCESS (OW)	68020 READ/WRITE (OW)
WRITE-BACK TRANSMIT BUFFER DESCRIPTOR PROCESS (OW)	WRITE-BACK RECEIVE BUFFER DESCRIPTOR PROCESS (OW)
FETCH RECEIVE ENTRY PROCESS (OW)	WRITE-BACK TRANSMIT BUFFER DESCRIPTOR PROCESS (OW)
FETCH TRANSMIT ENTRY PROCESS (OW)	FETCH RECEIVE ENTRY PROCESS (OW)
DATA MOVE PROCESS (HW)	FETCH TRANSMIT ENTRY PROCESS (OW)
68020 READ/WRITE (OW)	DATA MOVE PROCESS (HW)

### 3.1.3 XMI INTERFACE

The XMI interface is implemented in a 224 pin, surface mounted, 1.0 micron Sea-of-gates (channel-less) gate array known as the ESP. It interfaces three logic elements: the XMI (via the XCI bus and the XMI corner), the Packet memory (via the PBI bus) and the Adapter Manager subsystem (via the ESP bus).

#### 3.1.3.1 THE XCI BUS

Full details of the XCI bus may be obtained from the "Interfacing the XMI" chapter of the XMI specification

### 3.1.3.2 THE PBI BUS

The PBI bus is almost identical to the RMC bus (refer to the RMC Specification for full details). Following are details of the differences from the RMC specification.

#### 3.1.3.2.1 PBI BUS PROTOCOL

The PBI Interface consists of the exact same control and data signals that are present in the RMC Interface. The functions that the PBI signals provide, in terms of bus timing, is identical to that of the RMC. The only difference that exists between the two interfaces is found in the Command/ Address bit definitions.

Within the ESP gate array, there are two possible sources of PBI transactions, the Data Mover and the Adapter Manager. Depending on the source of the transaction, the Command/Address information will appear slightly different as seen in the table below.

**Table 2: PBI INTERFACE COMMAND/ADDRESS BIT DEFINITIONS FOR DATA MOVER**

Transaction	A31-A29	A28	A27	A26- A25	A24	A23-A2	A1	A0
Buffer Read	Burst Size	0	0	00	0	Address	0	1
Buffer Write	Burst Size	0	0	00	0	Address	0	0
Descriptor Read	000	0	0	01	0	Address	0	1
SDescriptor Read	001	0	0	01	0	Address	0	1
Descriptor Write	000	1	0	01	0	Address	0	0
Clear Own	000	0	0	11	0	Address	0	1

Note that in the above table there has been added a special buffer descriptor read operation specified as SDescriptor Read. This transaction allows the Data Mover to gather more information about the packet than just the buffer descriptor. There is a second longword of data in the transaction, the corresponding page table entry.

**Table 3: PBI INTERFACE COMMAND/ADDRESS BIT DEFINITIONS FOR ADAPTER MAN-  
 AGER**

Transaction	A31-A29	A28	A27	A26- A25	A24	A23-A2	A1	A0
Buffer Read	Burst Size	0	1	00	0	Address	1	x2
Buffer Write	Burst Size	0	1	00	0	Address	0	x2
Descriptor Read	000	0	1	01	0	Address	1	x2
Descriptor Write	000	x1	1	01	0	Address	0	x2
Clear Own	000	0	1	11	0	Address	0	x2
Special <sup>1</sup>	x3	1	x3	x3	1	Special Func- tion	x3	x2

<sup>1</sup>Special Cycles are R/W PTE's and R/W CSR's

- x1 = 1 implied Clear Own cycle, ie, PMC does automatic clear own  
= 0 explicit Clear Own Cycle will follow.
- x2 = 0 if the current transaction is Host to PMC  
1 if the current transaction is PMC to Host
- x3 = this field is transaction type dependent.

### 3.1.3.3 THE ESP BUS

This bus will be described in detail in the Adapter Manager subsystem section of this chapter.

### 3.1.4 THE ESP GATE ARRAY

The above interfaces form the communication mechanisms that are required for the ESP to perform it's various functions. The ESP will perform the following functions :

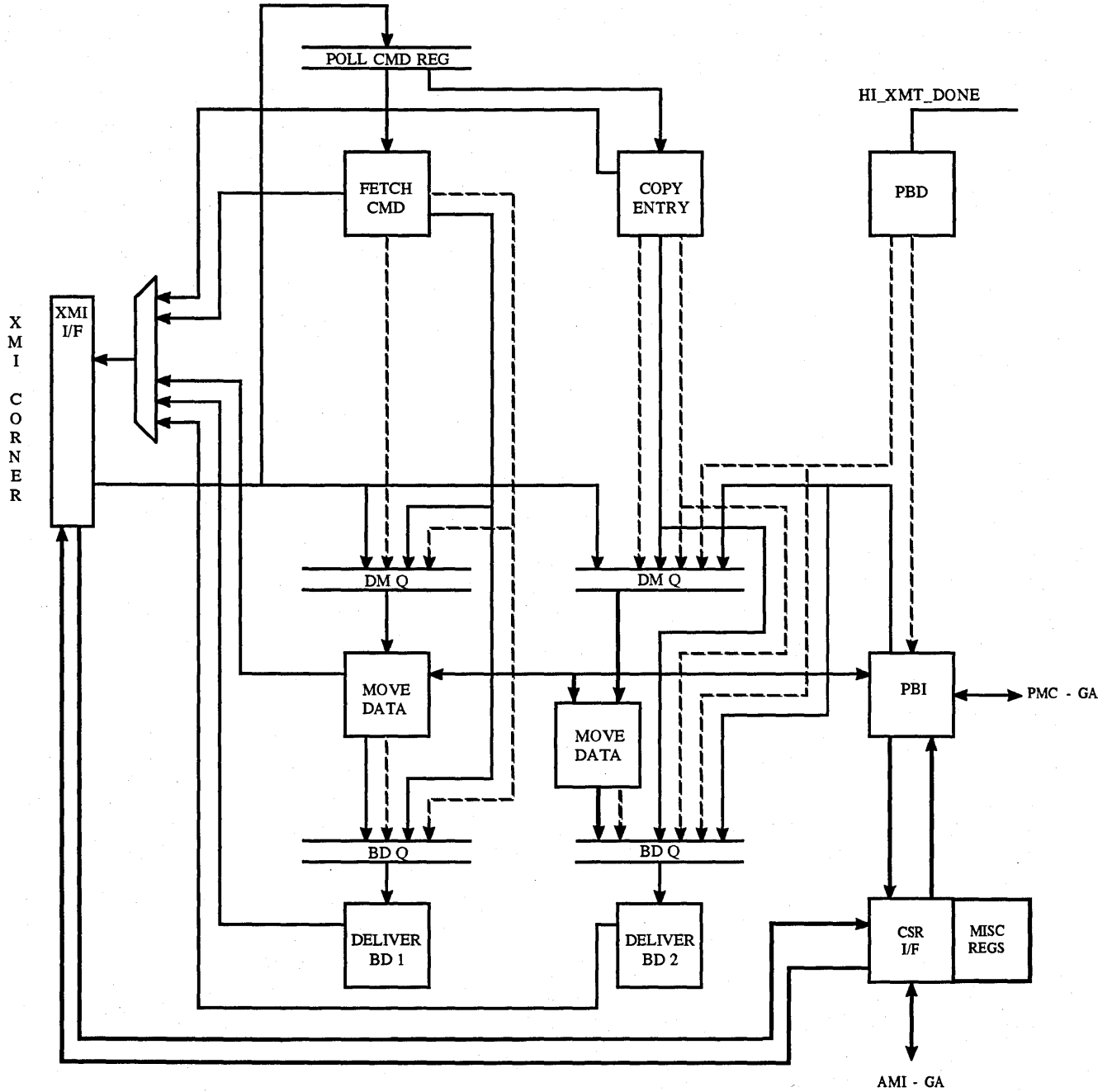
- Prefetch Host Transmit Ring Entries. An independent function within the ESP, this process understands the structure of the Host transmit ring and stores up to an octaword of information. As data is read by this process, it is decoded and passed on to the DM process for execution.
- Prefetch Host Receive Ring Entries. An independent function within the ESP, this process understands the structure of the Host receive ring and stores up to an octaword of information. As data is read by this process, it is decoded and passed on to the DM process for execution.
- Prefetch Packet Buffer Descriptors to be used in data transmissions from the Adapter to Host. An independent function within the ESP, this process understands the structure of the PMC Host Transmit Ring and stores up to one Buffer Descriptor. As data is read by this process, it is decoded and passed on to the DM process for execution.
- Change Host Transmit Ring Entry ownership and update status. An independent function within the ESP, this process changes the ownership of an entry and supplies transmission status at the completion of a data movement from the Host memory to Adapter packet memory. This process can store information for up to two transmit packets.
- Change Host Receive Ring Entry ownership and update status. An independent function within the ESP, this process changes the ownership of an entry and supplies reception status at the completion of a data movement from Adapter packet memory to the Host memory. This process can store information for up to two receive packets.
- Data Movement (DM process) of data from either Host to Adapter or Adapter to Host. An independent function within the ESP, this process compiles the information received from other processes within the ESP and performs the particular data movement called for. The DM process can store enough information to move up to two full or partial transmit packets and up to two full or partial receive packets. Priority for direction of packet transfers is programmable, with high priority defaulted to the movement of data from Adapter to Host. This process contains a dual hexaword buffer for continuous data transfers. All other functions compile information for or get information from the DM process and strive to stay ahead of this process so that a data movement will be present at all times, if possible.

The functions described above form the bulk of the logic complexity within the ESP gate array. These functions blend together to create a pipeline structure for the movement of data in and out of the ESP. Each function can be thought of as an element within the pipe, allowed to work independently on whatever operation it must perform. The functions communicate with each other by use of common data structures and valid bits. As an operation in one function is completed, the results are passed to next function by loading the results into the next functions data structure and thereby setting a valid bit.

The ESP handles other functions besides the ones described above. These functions are as follows:

- Provide a means for the Adapter Micro-processor to access the Adapter packet memory and the Host memory. The 68020, through use of the CSR interface can direct the ESP to perform up to an octaword XMI read or write transaction and up to a octaword PBI transaction.
- Maintain a set of XMI and Adapter specific registers.

Figure 6: ESP FUNCTIONAL BLOCK DIAGRAM



XFA\_HI\_ALT\_1\_NO\_XLATE.UIS

## **3.2 ADAPTER MANAGER SUBSYSTEM**

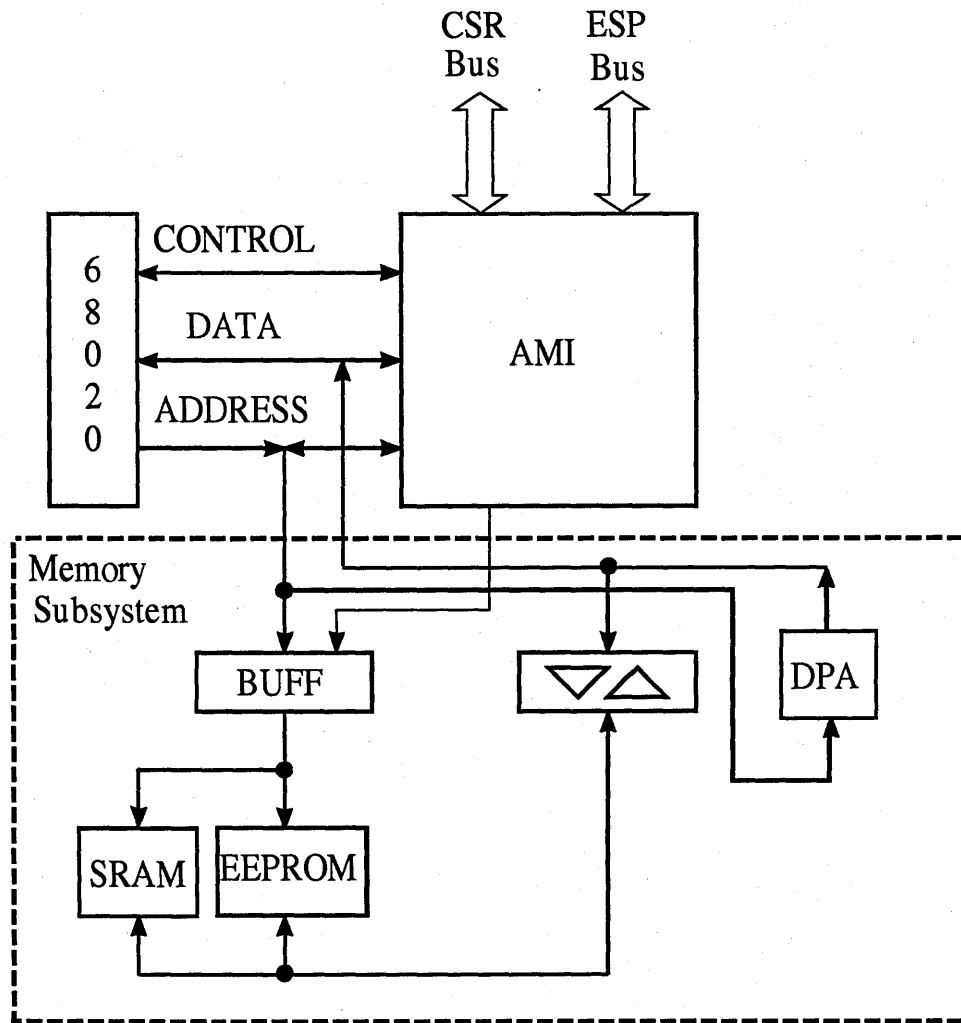
### **3.2.1 OVERVIEW**

The Adapter Manager will act as a Node Processor executing code to perform the following tasks:

- Self Test
- SMT Frame Processing ( SMT/CMT )
  - FDDI Physical Initialization & Control
  - FDDI Logical Initialization & Control
- XID and TEST Processing
- Remote Console Frame Processing ( MOP )
- Loopback ( MOP )
- Adapter Control via Port Driver Commands

The Adapter Manager will be a processing entity based on the Motorola MC68020 32-bit Microprocessor.

Figure 7: 68020 SUBSYSTEM - BLOCK DIAGRAM



### 3.2.2 AMI INTERFACES

The AMI has two interfaces. One is the CSR bus which is the Node Processor bus described in the RMC chip specification. The second is the ESP Bus.



### 3.2.2.1 THE ESP BUS

The ESP bus has a simple protocol. The AMI is always master, the ESP is always the slave. Data and address are dealt separately. The AMI can address up to 64, 32-bit registers within the ESP. There is a dedicated read/write line. Two of the ESP registers have a special function in that, on writing them, the ESP initiates either an XMI transaction or a PMC transaction. While such a transaction is in effect, the ESP asserts a corresponding busy signal. For Full details, refer to the AMI design specification. Following are sample timings of a read and a write cycle.

Figure 8: ESP BUS READ

### AMI READ from ESP Register

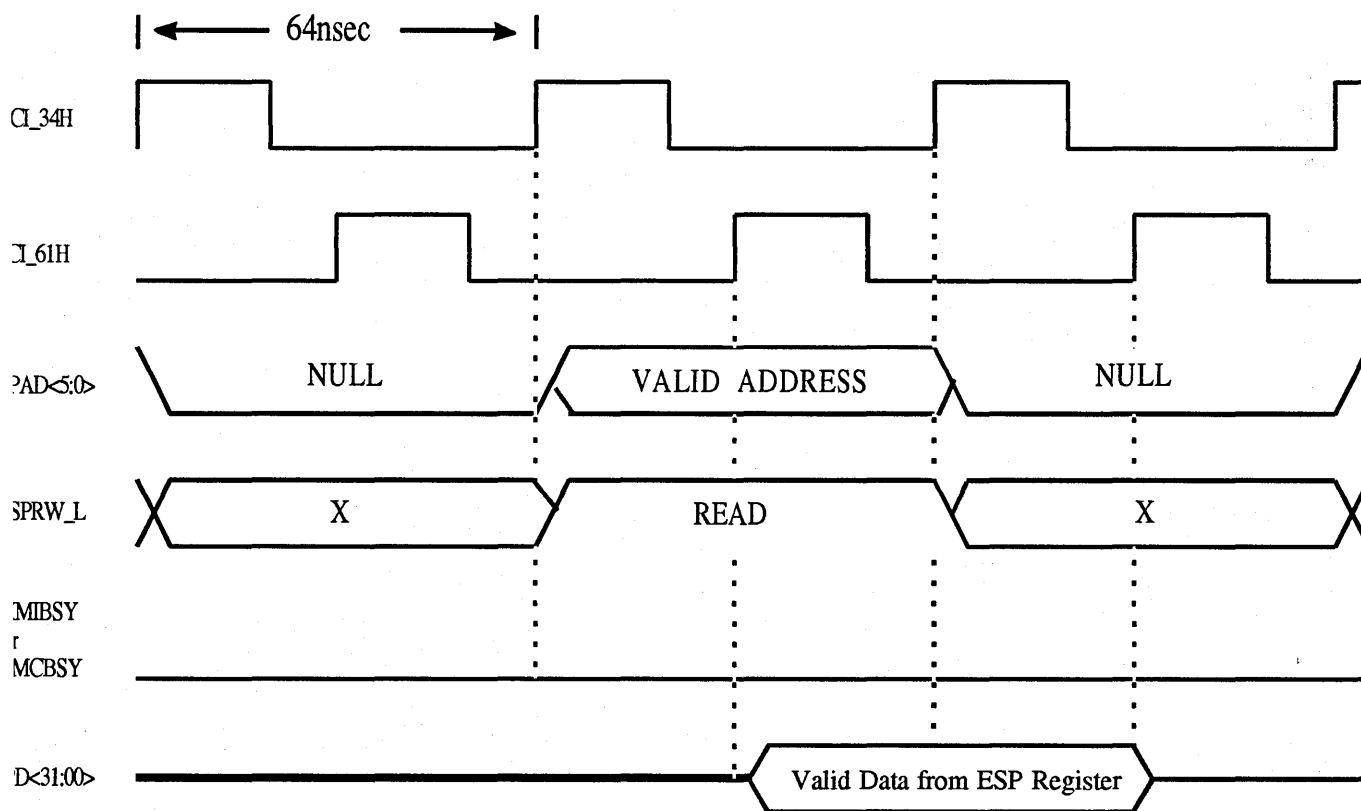
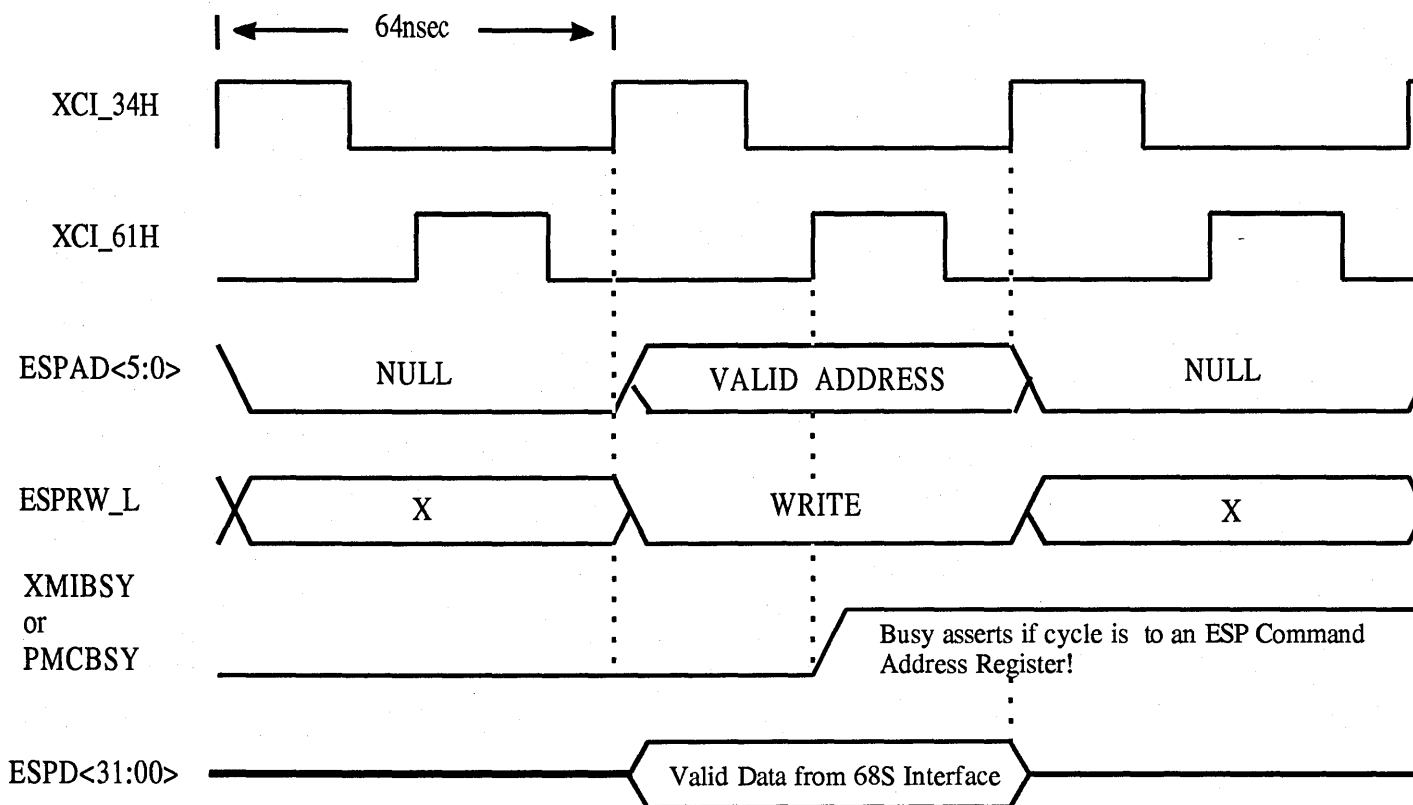


Figure 9: ESP BUS WRITE

### AMI WRITE to ESP Register



### 3.2.3 MICROPROCESSOR

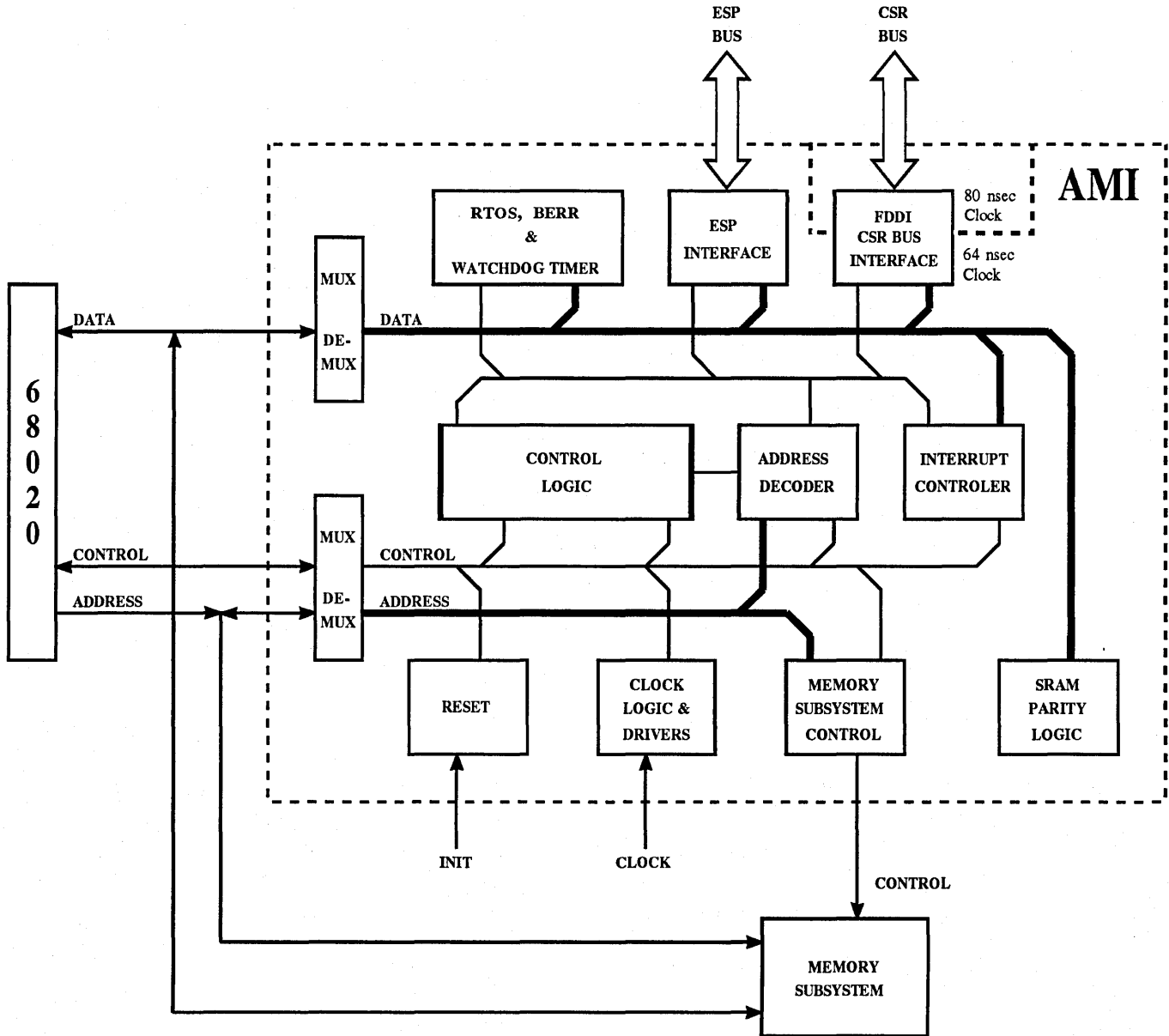
The microprocessor chosen for the Adapter Manager is a 16.67Mhz Motorola MC68020. The MC68020 will be run on a 50% Duty Cycle 15.625Mhz ( 64nsec ) clock derived from the XMI Corner XCI 34H Clock and circuitry within the AMI. Single stepping the microprocessor will not be supported in hardware, but can be achieved with an In Circuit Emulator. The MC68020 is a high performance microprocessor implemented in HCMOS technology and contains 32-bit registers and data paths, 32-bit physical addressing, on chip cache, dynamic

bus sizing and a pipelined architecture. At 15.625Mhz the 68020 offers approximately 2 MIPs performance. The device used in this implementation is packaged in a 114 Pin PGA.

### 3.2.4 AMI GATE ARRAY

The AMI Gate Array will combine all the necessary function units and logic, with the exception of memory, to support the 68020 microprocessor.

Figure 10: AMI BLOCK DIAGRAM



### 3.2.4.1 FUNCTIONAL UNITS

- **Address Decoder:** Physical addresses produced by the 68020 are decoded in this unit to generate the enable signals for memory and peripheral devices within the Gate Array. This function unit will also support the control signals for dynamic bus sizing.
- **CONTROL Logic:** This logic in co-ordination with the Decode Logic will provide the 68020 with the DSACK signals required to notify the 68020 that a data transfer operation is complete. DSACK signals must be generated during interrupt acknowledge cycles as well. This Data Acknowledge function block will also contain a wait state generator to add machine cycles for slow peripheral accesses such as to EEPROM.
- **RTOS, BERR and WATCHDOG Timer:** This function unit will contain the RTOS Timer that provides the periodic interrupt for the RTOS Real Time Clock. Estimates on the periodicity of the timer output to be programmed are in the range of 5msec to 10msec. The RTOS timer output will be programmable to a maximum period of 1 second. The RTOS Timer is programmed with a period of 10 milliseconds.

A Bus Cycle Timer will be included in this section to provide a Bus Error signal to the 68020 if a bus cycle does not complete in a specified period of time. Upon receipt of a Bus Error the 68020 will trap to a error handling routine. This is to prevent a "hung" 68020 bus in the event of an error. The Bus Error Counter can be programmed for up to 4 milliseconds before expiring if no DSACKL is received.

The Watchdog Timer will provide 'a keep alive' function for the 68020. The Watchdog Timer is periodically cleared by a 68020 CSR access under firmware control. In the event that firmware branches off in an uncontrolled manner ( failure ), the Watchdog Timer will time out and generate an interrupt. The interrupt will cause the 68020 to vector to an Error Handling Routine. In addition, dedicated error reporting signals XERCOD<1:0> between the AMI and the ESP will be asserted with an error code. This error reporting mechanism is provided in cases where the 68020 unable to report errors. The assertion of these signals will notify the Host via the ESP that a Node Reset is requested. The Watchdog Timer can be programmed for up to 274 seconds before it expires if no Watchdog Counter Clear strobe is detected. The Watchdog Timer is programmed with a period of 10 seconds.

- **FDDI Chips CSR Bus Interface:** This is an interface to the ELM, MAC, MAC\_CAM, RMC and PARSER CSR Bus. The bus interface allows the MC68020 to perform read/write accesses of the Control/Status Registers contained within these units. A level of re-synchronization from the 68020's 64nsec clock to the FDDI Chips BYTCLK (80nsec) will occur such that the interface is in synch with the FDDI time domain. The CSR Bus is a 16-bit bi-directional data bus and will be justified to the 68020's "low word" or D31 thru D16. Byte parity generation will be provided to the PARSER only.
- **ESP Interface:** This is an interface channel between the AMI Gate Array and the ESP Gate Array. This interface will provide access to the ESP/Host as well as PMC/ Packet Buffer Memory. The path to the Host is used for accessing the Host Command Ring and Unsolicited Ring. The path to the PMC is used for accessing PMC Registers and Packet Memory. Both paths are indirect methods of communications. The ESP Bus Interface supports generating and checking longword parity. Separate parity logic is contained within this section.

- **Interrupt Controller:** All the interrupts to the 68020 come into this function unit and are prioritized before connecting to the 68020. The 68020 has 7 IPLs ( Interrupt Priority Levels ), with level 7 being the highest level and non-maskable. Incoming interrupts will be synchronized to the 64nsec clock. Each interrupt will have associated with it a unique vector, vector register contents will be programmable. Separate Interrupt Mask Bits settable via CSRs will be provided as a hardware mask for interrupts. Separate Interrupt Enable Bits settable via CSRs will be provided as individual hardware enables for interrupts.

**Table 4: INTERRUPT STRUCTURE**

IPL	DESCRIPTION	PRIORITIZED SOURCE
7	System Error,RTOS	Watchdog Timer AC Power Fail AMI/ESP Bus Parity Error (AMI Read operation)
6	DEMFA	ESP Chip FDDI CSR Bus Parity Error
5	FDDI Interface	ELM PARSER
4	FDDI Interface	MAC RES RMC SMT Priority
3	DEMFA	PM Start Transmit
2	System Required	Hardware Timer
1	RTOS Required	Event Level Monitor

- **Reset Logic:** The reset logic provides a path to reset the entire 68020 subsystem. Inputs to the Reset Logic are INIT\_L from the ESP G.A. and SRAM\_PE. The INIT\_L signal from the ESP G.A. can be asserted due to: Power Cycle, Backplane Reset or Node Reset ( Device Driver Generated ). The manner in which the Watchdog Timer resets the Adapter Manager is by requesting a Node Reset from the Host via the error bits XERCOD[1:0]H that are a direct connect to ESP.

The XMI signal AC\_LO L will be used by a portion of this logic to indicate the difference between a Cold and Warm Reset. AC\_LO L does not get asserted on a Warm Reset, but only on a cold power-up Reset. The Parity Error Address Register contents are maintained through a Warm Reset. The Parity Error Address Register is only initialized on a Cold Reset by a signal derived from AC\_LO L.

When an SRAM Parity Error occurs, logic will put the 68020 into a Reset State and not remove it until INIT\_L cycles.

A Reset Disable Bit is provided in the GPCSR to prevent the INITL line from resetting the AMI during a warm reset. This allows the 68020 to generate a reset of the adapter without losing its code. The bit is automatically cleared after the INITL line is cycled once.

- **Clock Logic & Drivers:** Logic is provided to create a 50% duty cycle 68020 Clock from the 33% Duty Cycle CLK34H. Internal clocks are buffered and their loading is balanced as much as possible to minimize clock skew throughout the Gate Array.
- **Parity Logic:** This logic generates and checks SRAM Data Bus Parity. The Byte Parity generated defaults to ODD Parity. Provisions are made to generate EVEN Parity to facilitate testing of the Parity circuit. A parity error indication is obtained only on an SRAM read cycle.

### 3.2.5 68020 MEMORY SUBSYSTEM

Contained in this subsystem will be the local memory for the MC68020. The subsystem would include: SRAM, EEPROM and the Default Physical Address PROM. Buffered versions of the 68020 address and data bus will be connected to the Memory Subsystem data path to provide added drive capability for interfacing to the memory array. The address and lines to the Memory Subsystem will be buffered externally from the AMI Gate Array providing added drive capability.

- **SRAM:** The SRAM will be used for operand loads and stores as well as program space during program execution. The SRAM will be configured to provide a density of 256KByte of zero wait state memory. Byte Parity will be provided with 4 bits of Parity Protection for the SRAM data bus.
- **EEPROM:** EEPROM will be provided to contain code for: Kernel Code, Self Test, Initialization, RTOS Monitor, SMT, CMT, Port Control, XID/TEST, Local and Remote Management, etc. The contents of EEPROM will be transferred over to SRAM after diagnostics have been run on the SRAM. This transfer of memory contents will be performed by the 68020. This will allow code to run with zero wait states.

EEPROM will be configured to provide a density of 256KByte and will be accessed via longwords only. Each longword read from EEPROM requires 4 wait states due to the access time of the EEPROM ( 250nsec ). A special firmware timed operation is required when writing to EEPROM due to the special nature of the devices. A 1 microsecond delay must be inserted between page mode writes and a 10 millisecond delay must be inserted between different page write accesses while contents are being moved within the EEPROMs.

- **DEFAULT PHYSICAL ADDRESS PROM:** The Default Physical Address ( Node Address ) PROM will contain six bytes of FDDI Node Address unique to each adapter. The 68020 will have ability to read the entire 32 Bytes within the PROM. DPA PROM data integrity will be protected by Checksum.

### 3.2.6 SUBSYSTEM DATA INTEGRITY

Parity will be used to provide SRAM Data Integrity Protection as well as AMI/ESP Bus Protection. EEPROM contents integrity will be confirmed with several Checksum calculations and compares. Separate checksums will be calculated for each major block of code, such as Self Test Code, Functional Code, etc. Byte Parity will be provided on the SRAM Data Bus and Longword Parity will be provided on all transactions to the AMI/ESP Bus. Byte Parity will be generated for writes to the FDDI CSR Bus on accesses to the Parser only.

### 3.2.7 POWER UP & SELF TEST

- **Power-Up Reset:** The ESP Chip will provide an INIT\_L Signal to the AMI. Upon de-assertion of the Reset Signal, the 68020 will obtain its Stack Pointer from Address \$00000000 and its Program Counter from Address \$00000004 in EEPROM and begin executing Start Up and Self Test Code.

Test Logic within the AMI will be provided only as needed to ease the task of self test. The AMI will rely mainly on the 68020 performing the testing of the AMI Gate Array. The 68020 will run Self Test Code out of EEPROM until it has sufficiently tested enough SRAM to enable it to safely run Self Test Code out of SRAM. Self Test Code in EEPROM will be copied over to SRAM by the 68020. Executing code out of SRAM will minimize self test execution time. Self Test execution time is critical due to the 10 second maximum time imposed by the XMI Bus.

- **Error Status:** Two dedicated lines from the AMI will go to the ESP and PMC Gate Arrays. These lines, XERCOD[1:0], will provide 4 possible codes that represent the error status of the AMI. The status conditions are: Null, Watchdog Timer Expired, SRAM Parity Error and ESP Bus Parity Error.
- **Loading Functional Code:** Once Self Test is complete, the 68020 will move the Functional Code from EEPROM to SRAM. Upon completion of this, the 68020 will execute other initialization procedures to transition to Uninitialized State and await a command from the Host.

### 3.3 PMC GATE ARRAY

The Packet Buffer Memory subsystem interfaces with the ESP and RMC gate arrays via the PBI bus and RMC bus respectively. Refer to the RMC specification for full details of these busses and see the Host Interface section of this chapter for differences specific to the PBI bus.

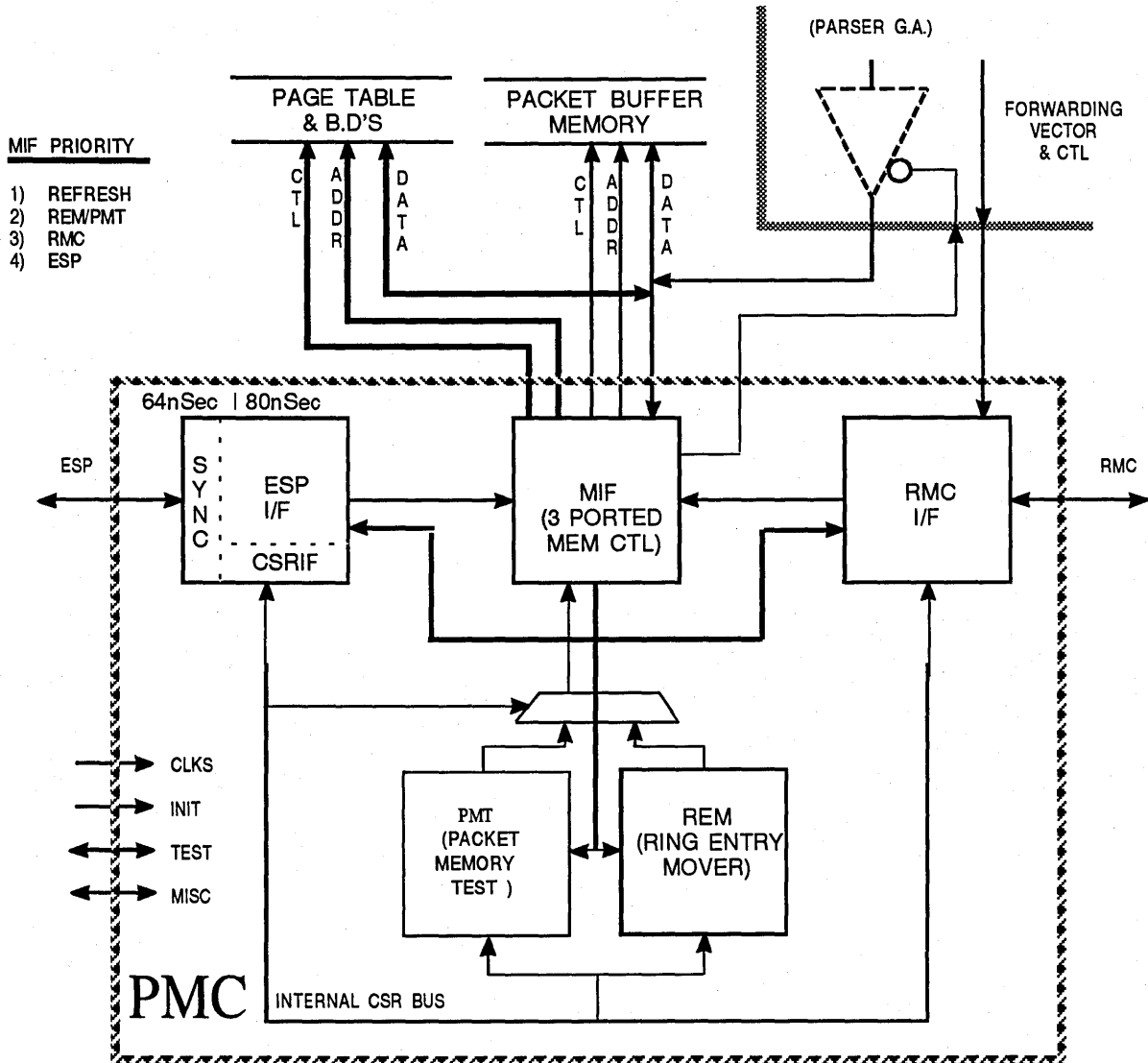
#### 3.3.1 OVERVIEW

The Packet Memory Controller (PMC) is a logical four ported and physically three ported device with two external ports (ESP, RMC ) and one internal port (Ring Entry Mover). It's function is to regulate the reading and writing of the Packet Memory as required by the demands of the various interfaces. The processes of control include arbitrating access between users of the Packet Buffer and "moving" ring entries by means of manipulating the virtual address Page Table Entries.

The logical four ports of interface are the Host, Adapter Manager, RMC and Ring Entry Mover. There is also an interface with the Parser to enable the forwarding vector into the memory system.

The clock of the controller will be synchronized to the 80nSec FDDI clock.

Figure 11: PACKET MEMORY CONTROLLER - FUNCTIONAL BLOCK DIAGRAM



PMC - FUNCTIONAL BLOCK DIAGRAM

XFA\_PMC\_FBD.UIS



### 3.3.2 INTERFACES

#### 3.3.2.1 RMC INTERFACE

The RMC (Ring Memory Controller) interface is the port used by the RMC for moving FDDI packets between the packet buffer and the FDDI ring. For full details of the RMC Bus, refer to the RMC Functional Specification.

#### 3.3.2.2 ESP INTERFACE

The ESP interface is the port used by the ESP gate array for moving FDDI packets between host memory and the Packet Buffer. This interface is also used by the Adapter Manager; the ESP multiplexes host data and Adapter Manager data over the ESP/PMC interface bus. The interface is almost identical to the RMC interface.

### 3.3.3 FUNCTIONAL BLOCK DESCRIPTIONS

#### 3.3.3.1 PACKET BUFFER MEMORY

The Packet Buffer Memory consists of 1024KBytes DRAM (expandable to 4 MB given next generation memory) implemented as a single physical bank of memory addressable by longword only. Data protection is by the use of byte parity.

#### 3.3.3.2 PAGE TABLE AND BUFFER DESCRIPTOR MEMORY

The Page Table and B.D. Memory is 16K Longwords (expandable to 64K) of SRAM divided into specific allocations as follow: the Page Table, 14K (32K) longwords and the Buffer Descriptors 2k (8K) longwords. This allows the Packet Buffer to be mapped seven (four) times, ie, virtual space is seven (four) times physical space. (Numbers in parenthesis indicate that for a 4MB Packet Memory). This memory block also contains the Forwarding Vector and 'color' of each page.

Color is a 2\_bit field in a Page Table Entry. Color indicates to which receive ring the buffer (page) was initially assigned. When buffers become free they are returned to the ring of initial assignment. Color value has the following meaning:

---

**Table 5: COLOR AND RING ASSIGNMENTS**

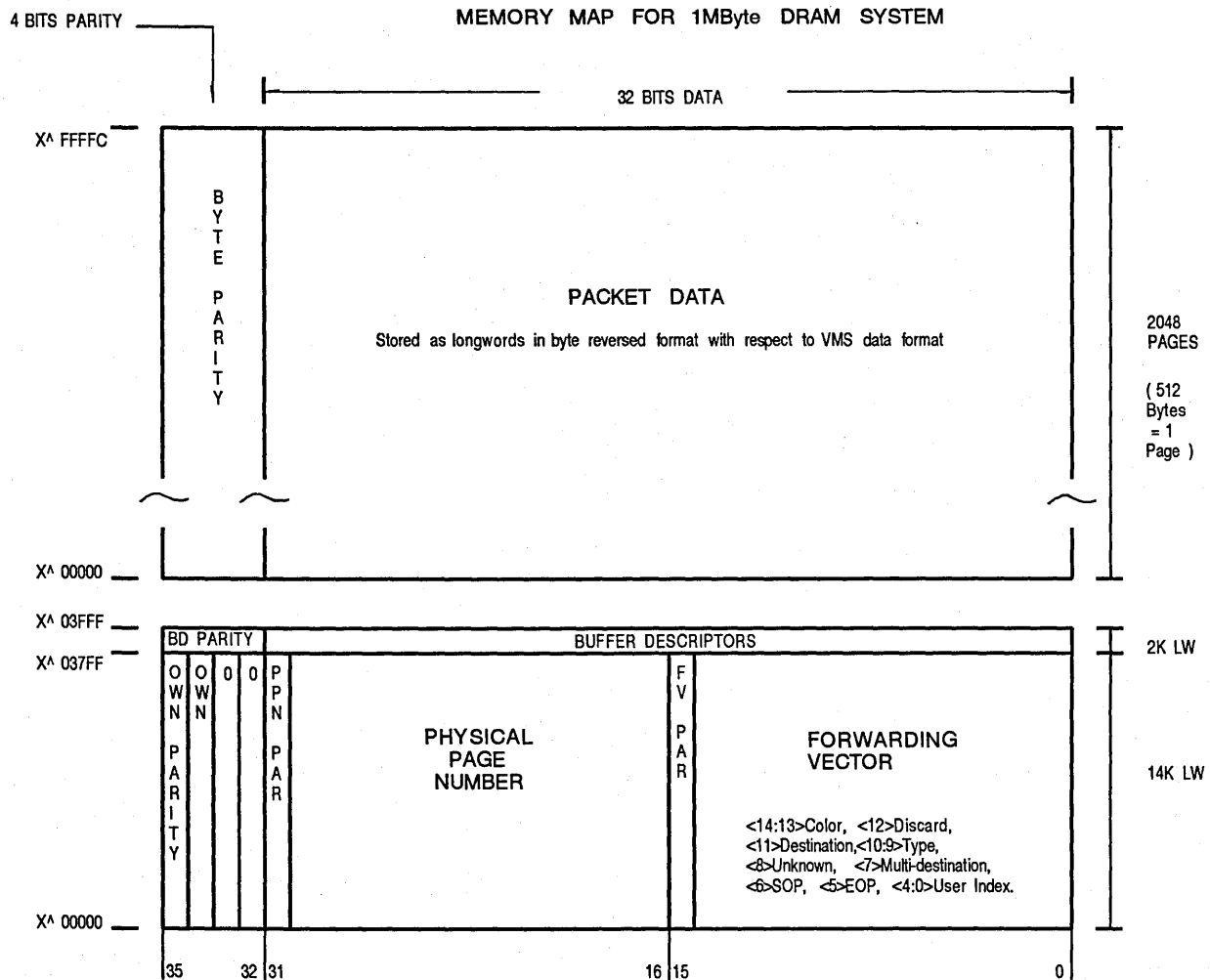
---

COLOR VALUE	Assigned Ring
^B00	Unassigned
^B01	RMC Receive Ring
^B10	AM Receive Ring
^B11	Host Receive Ring

---

Data protection is by means of parity. The following fields are independently protected by a single bit of parity: the Physical Page number, the Forwarding Vector and the Ownership field. The Buffer Descriptors are protected by byte parity.

Figure 12: MEMORY ALLOCATION



XFA\_PMC\_PMA.UIS

### 3.3.3.3 RING ENTRY MOVER

The Ring Entry Mover (REM) has inherent knowledge of the three external logical ports, being RMC, Adapter Manager and ESP. It interfaces with them via three pairs of rings of the form described in the RMC Functional Specification. It receives notification of all received and transmitted packets based on transmit and receive done status from the three external ports. When a packet is queued by the REM to a transmit queue, notification of the event is passed to the appropriate logical interface. The base address and length of all rings to be controlled are transferred to the REM by the Adapter Manager (via the ESP) during initialization. The REM 'moves' packets by manipulating the Page Table Entries based on information contained in a forwarding vector which is a part of the Page Table Entry.

### 3.3.3.4 PMT - Packet Memory Test

The packet memory test block is a functional block which, on power up or other initialization, will automatically test the DRAM associated with the PMC. The test can be disabled after power up such that a programmed initialization will not enable the test. The interface to the memory test is programmable in order to enable debug and manufacturing.

### 3.3.3.5 MIF - MEMORY CONTROLLER

The Memory Interface is a three port Memory Controller. There is a 'hidden' forth port, being the refresh for the DRAM. The MIF automatically takes care of the refreshing of the DRAM. Access is prioritized in the following manner:

1. Refresh
2. REM (or PMT)
3. RMC
4. ESP

When a memory access opportunity is present, the requester of the highest priority is granted. By design, the ESP cannot normally be locked out. However, when the Packet Memory Test is running, it will lock out all other traffic to the Rams except refresh. Otherwise worst case lockout for the ESP is approximately 200  $\mu$ Sec but is typically less than 1uSecs.

### 3.3.3.6 RMC I/F

The RMC I/F interprets RMC Instructions and translates them into a form acceptable to the memory system. The RMC instructions are strictly as defined in the RMC Functional Specification. This interface is 80nSec synchronous.

The RMC I/F will, on an RMC Receive Clear\_Own cycle, co-ordinate the enabling of the forwarding vector onto the Packet Buffer Memory Data bus so that the vector is written to memory as an integral part of clearing the OWN bit.

### 3.3.3.7 ESP I/F

The ESP I/F is the hardware port for both the Host and the Adapter Manager This interface synchronizes the 64nsec of the ESP to the 80nSecs of the Gate Array. It supports a superset of the RMC instructions. The differences from the RMC are as follow:

1. A read Buffer Descriptor by the Host interface for transmit (Packet Memory to Host) will result in two data cycles, not one, the first being the page table entry and the second being the Buffer Descriptor.
2. Additional commands for the Adapter Manager:
  - CSR access
  - R/W Page Table Entry

### 3.3.4 TRANSMIT/RECEIVE DATA STREAM PRIORITY

The PMC implements logic which controls the priority of the Transmit and Receive Data streams at times when there is contention.

Priority is generated in the RMC interface and is used by the Ring Entry Mover. If the RMC is transmitting, the REM will prioritize packet moves to the RMC Transmit queue.

The definition of this algorithms is contained in the PMC Design Specification.

### 3.3.5 ADDITIONAL FEATURES

- Address translation may be enabled/disabled. This is to allow for page table entry initialization and for debug purposes.
- The REM may be gracefully 'frozen' (after completing the current buffer or packet move).
- All HOST\_RCV packets may be forced to the Adapter Manager (by default, all packets received from the host are for the RMC).
- All RMC\_RCV packets may be forced to the Host (for debug purposes).
- All RMC\_RCV packets may be forced to the Adapter Manager (for debug purposes).
- In order to complement the RMC's ability to byte swap, the ESP interface can recognize the host data stream (as opposed to the Adapter Manager's data stream) and can 'unswap' the data and deliver it to the host in the format required for the VAX and MIPS based hosts. This enables data to be delivered to the Adapter Manager in byte swap mode and so help compatibility with SMT (68020) firmware.

## 3.4 THE FDDI CHIP SET

The DEMFA Adapter utilizes the NAC designed FDDI chips being, ELM, MAC and RMC. The knowledge of ANSI's FDDI specification is assumed.

The chips perform the following functions:

#### *ELM (Elasticity Buffer and Link Management)*

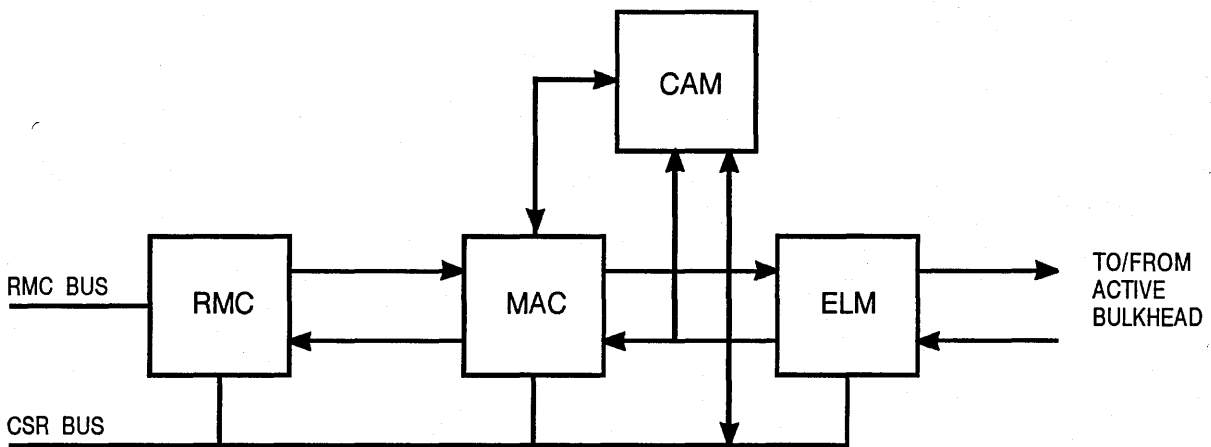
- Elasticity Buffer (for synchronizing recovered and local clocks)
- 4B/5B encoding/decoding.
- Line State Detection
- Repeat filter for MAC-less connection
- Physical Connection Management state machine

#### *MAC (Media Access Control)*

Implements ANSI's FDDI MAC standard. Major features:

- Respond to the Frame Control byte for received frames
- Destination Address filtering

Figure 13: FDDI CHIP SET



FDDI\_CHIP\_SET.UIS

- Perform CRC checking for receive packets
- Save frame status indicators
- Keep a count of the good and bad frames received
- Assemble packets for transmission
- Optionally generate CRC for transmit packets

- Can internally generate and send Claim, Beacon and Void Frames

### *CAM (Contents Addressable Memory)*

Compatible with MAC chip. Major features:

- Holds 64 of 48-bit addresses
- 16-bit interface for Node Processor
- Can be set in read, write, clear and match mode
- Match can be performed both from MAC and Node Processor

### *RMC (Ring Memory Controller)*

While not an FDDI chip, it is an integral part of DEC's FDDI implementation. Following are some of the major features:

- Independent transmit and receive channels
- Fixed size (512 Byte) buffers
- Fragment filter to discard small 'non-packets'
- 4 or 8 longword burst mode to packet memory
- Circular (ring) buffer support.
- Buffer 'ownership' to pass packet memory access rights.

## **3.5 PARSER SUBSYSTEM**

### **3.5.1 OVERVIEW**

The Parser Subsystem is required to do the filtering based on Frame Control(FC), Destination Address(DA), LLC header (802.2), and Buffer Descriptor during the receive of a packet from RMC. The Figure 14 shows the logical place of the Parser Subsystem in the overall DEMFA. As shown in the diagram, the Parser extracts the Packet header information from the RMC bus. The Parser then gets the filtering information from the Database (DB MEM) for that Packet header and finally it generates a Forwarding vector (see Figure 15) for each page of the packet and sends it to the Packet Memory Controller (PMC). The PMC then stores the Forwarding Vector in the page table entry corresponding to respective page(s) of the packet.

The Parser Database contains the list of all enabled FCs, DAs and LLCs (DSAPs and PIDs). It also contains the filtering information required for packet parsing based on various combinations of FC, DA and LLC. The Parsing Algorithm uses this information to decide the destination of the packet.

The following lists the various steps involved in the generation of the Forwarding Vector for a single page packet and a multiple page packet.

- Single Page Packet
  - The RMC fills page in packet memory. (The Parser extracts the header information.)

- The RMC writes buffer descriptor. ( The Parser extracts the SOP, EOP and RCC from the descriptor.)
- The Parser generates the Forwarding Vector.
- Multiple Page Packet
  - The RMC fills the first page in packet memory. (The Parser extracts the header information.)
  - The RMC writes buffer descriptor for first page . ( The Parser does nothing.)
  - The RMC fills the second page in packet memory.
  - The RMC writes buffer descriptor for second page . ( The Parser extracts the SOP and EOP from the descriptor.)
  - The Parser generates the Forwarding Vector.
  - The RMC fills the third page in packet memory.
  - The RMC writes buffer descriptor for third page . ( The Parser extracts the SOP and EOP from the descriptor.)
  - The Parser generates the Forwarding Vector.
  - **And So On for other pages.....**
  - The RMC fills the last page in packet memory.
  - The RMC writes buffer descriptor for last page . ( The Parser extracts the SOP and EOP from the descriptor.)
  - The Parser generates the Forwarding Vector.
  - The RMC write buffer descriptor for the first page . ( The Parser extracts the SOP,EOP and RCC from the descriptor.)
  - The Parser generates the final Forwarding Vector.

The Forwarding Vector contains some information for the Ring Entry Mover such as Discard packet, the eventual destination of packet (Adapter Manager or Host Interface), SOP, EOP etc. The rest of the information in the Forwarding Vector is provided for the port driver who eventually delivers the packet to specific user(s).

The following list briefly describes some of the important features of the Parser Subsystem.

- This Parser supports 31 different users and one Unknown User. Some of the examples of users are DECnet, LAVC, MOP etc.
- It supports all classes of Frame Control, but
  - The Parser Database should always be programmed to discard those values of FC for which the Frame Address bit indicates the 16 bit addresses.
  - The Parser Database should always be programmed to discard the Nonrestricted and Restricted Tokens.
- It supports 63 different Destination Addresses and one combination for all other Multicast DAs (AMC feature). The destination addresses contain the station's individual address, all other individual addresses enabled by users and all multicast addresses enabled by users. (The broadcast address is treated as any other multicast address.)

- It supports 63 different combinations of LLCs means DSAPs for Non SNAP SAP and PIDs for SNAP SAP packets and one combination for Unknown User.
- The Parser is capable of filtering a packet based on only FC, and/or combination of FC and DA, and/or combination of FC, DA and LLC.
- The Parser is also capable of filtering a packet based on the information provided by Buffer Descriptor (Receive Completion Code).
- The Parser is capable of sending any packet to either Host (Port Driver) or Adapter Manager(AM).
- The Parser is capable of filtering a packet within the time equivalent to the minimum size of that packet.



Figure 14: PARSER SUBSYSTEM IN DEMFA

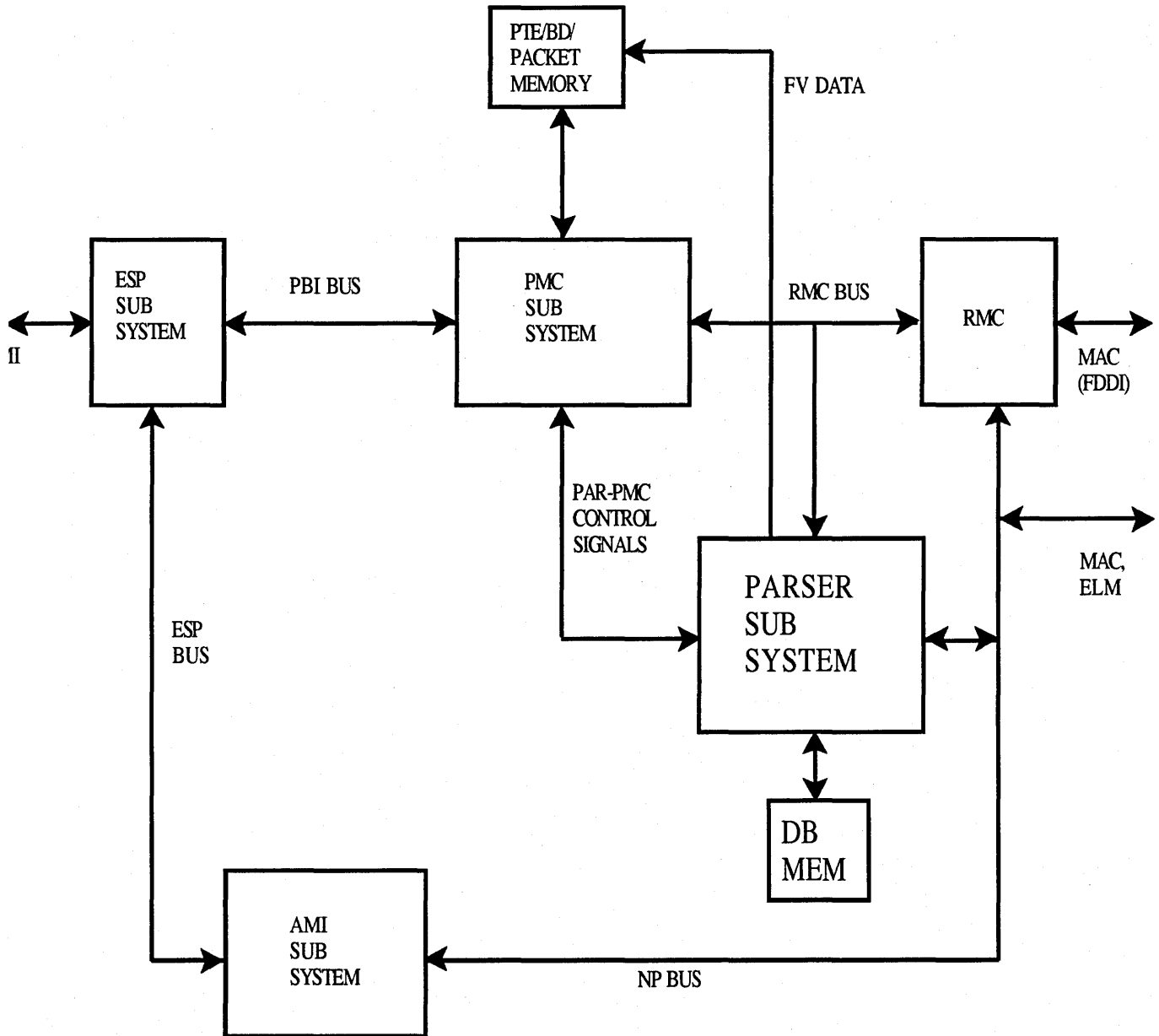
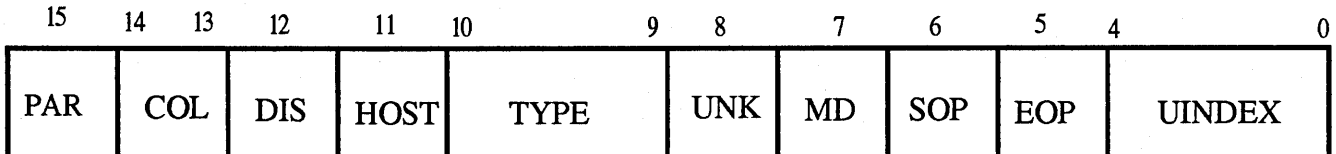


Figure 15: FORWARDING VECTOR



- PAR <15> - Parity
- COL <14:13> - This is the color field assigned to RMC.  
This field will always be 01.
- Dis <12> - Discard
- Host <11> - 1 Host  
- 0 Adapter Manager
- Type <10:9> - 00 XID/Test/Other  
- 01 SMT/MAC  
- 10 MOP Packets  
- 11 Error Packets

Note : This type field is valid only when the packet is destined for Adapter Manager.

- UNK <8> - The Packet is for the Unknown User
- MD <7> - There are multiple recipients for this packet.
- SOP <6> - Start of Packet
- EOP <5> - End of Packet
- UIndex <4:0> - User\_Index.

### **3.5.2 THE PARSER GA**

The Parser is being designed as an ASIC gate array with the 1.5  $\mu$  Sea-of-gates technology. The GA contains approximately 21K gates and it is housed in a 224 pin surface mount flat package. The following is a brief description of various interfaces contained within the Parser GA. Please refer to Figure 16 for this discussion.

#### **3.5.2.1 RMC INTERFACE**

The RMC Interface includes RMC Bus and RMC Control. This interface extracts the useful information from the RMC bus when RMC receives a new packet. It stores these information in an internal register file. The information extracted is Frame Control (FC), Destination Address (DA), Source Address (SA), and LLC header information such as DSAP, SSAP, Control and Protocol Identification (PID) and the Buffer Descriptor. The buffer descriptor provides information like SOP, EOP, and Receive Completion Code.

#### **3.5.2.2 ADAPTER MANAGER INTERFACE**

The Adapter Manager talks to Parser through this interface. The Adapter Manager interface contains a set of internal register file to store some important control and status information. This interface also contains registers to update Parser Database (RAM) and provides proper handshake information to RAM interface. This interface notifies the Adapter Manager of any error condition via an interrupt.

#### **3.5.2.3 RAM INTERFACE**

This interface performs two different tasks - It updates the Parser Database whenever the Adapter Manager needs it to be done and secondly it performs the actual FC, Destination Address and LLC matching. The results of this matching as well as other information from database are passed to the Algorithm State Machine which implements the parsing algorithm (See Appendix D) required for complete filtering.

#### **3.5.2.4 PACKET MEMORY CONTROLLER INTERFACE**

The Packet Memory Controller Interface communicates with Packet Memory Controller (PMC) with some handshaking signals. This interface is also responsible to provide the Forwarding Vector(s) to the PMC.

#### **3.5.2.5 ALGORITHM STATE MACHINE**

As the name indicates, this Finite State Machine (FSM) implements the parsing algorithm. (See Appendix D).

#### **3.5.2.6 BUFFER DESCRIPTOR DECODE STATE MACHINE**

This FSM performs two basic functions : i) It copies the SOP and EOP from the RMC interface to the Forwarding Vector, and ii) It decodes the Receive Completion Code and provides information to the Algorithm FSM which then completes the parsing algorithm.

### **3.5.2.7 SAP DECODE STATE MACHINE**

This FSM decodes the DSAP, SSAP and CNTL fields of the incoming packet and provides its results to Algorithm FSM which then completes the parsing algorithm.

### **3.5.2.8 TEST INTERFACE**

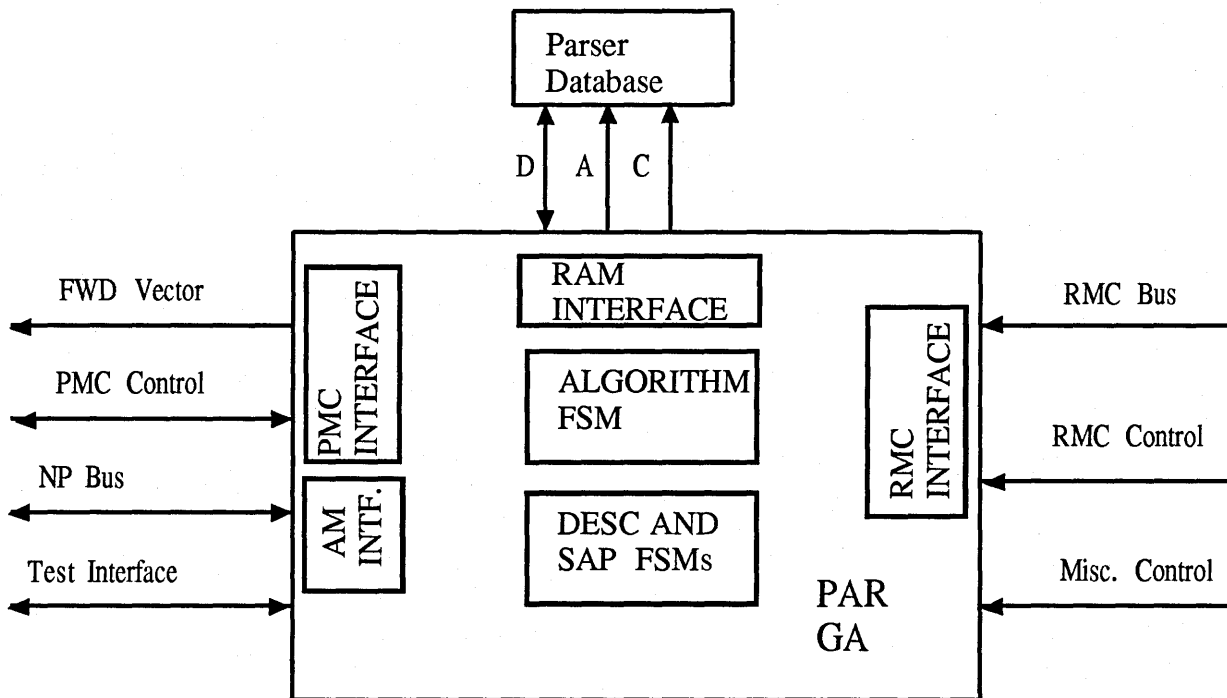
This interface implements all logic required to do the internal testing of the chip. Currently, it is planned to implement Boundary Scan and Parametric test.

### **3.5.3 THE PARSER DATABASE**

The Parser Database contains all the information required to do the packet filtering. It contains the list for all valid FCs, all enabled DAs and LLCs. In addition to these lists, it also contains the forwarding information about all types of packets as to their destinations, their types, their user indexes.

The Database is contained in Static RAM which is organized as 8KX72. The Database is protected by the byte parity, which is provided by the Adapter Manager while initializing and updating this Database.

Figure 16: PARSER SUBSYSTEM BLOCK DIAGRAM



D = Data  
C = Control  
A = Address

### 3.6 EXPLANATION OF REGISTER DESCRIPTIONS

In the following chapters there are descriptions of all the addressable elements within the DEMFA. Following are explanations of the conventions and abbreviations used.

- All registers are named and assigned a mnemonic.
- Access, ie XMI and/or Adapter Manager, is stated. Abbreviations used are:
  - R/W - This bit(s) is simple read and write
  - W1C - Write 1 to clear; this is a toggle bit(s), ie, writing with a 1 will toggle the value of this bit. Writing with 0 has no effect
  - RO - Read Only; this bit(s) is read only, writing has no effect
  - WO - Write Only; this bit(s) is write only, the value read is defined for each instance in the register definitions
  - SC - Special Case; the access of this bit(s) is defined in the bit(s) definition
- Addresses from either or both interfaces are given in hexadecimal. Addresses are absolute or relative (to a base address). See each register, or group of registers, for definition.
- Where appropriate, bits are assigned mnemonics.
- All register initialization values are given, either all bits = 0, all bits = 1 or refer to the individual bit definitions.
- All used bits are defined. The definition describes what the bit(s) means and what is its effect. Abbreviations used are:
  - MBZ - Must Be Zero; this bit(s) must be written as 0
  - RESV - Reserved; this bit(s) is not used.
  - NI0 - Not Implemented, 0; this bit is not implemented and is read as 0.
  - NI1 - Not Implemented, 1; this bit is not implemented and is read as 1.
- Basically any hardware detected error is fatal, with the exception of an RMC XMT detected bus data parity error. There is also a class of yet-to-be-determined firmware detected errors that may also be fatal. Fatal means that recovery can only be attempted by reinitialization.

# CHAPTER 4

## ESP REGISTERS

This chapter describes all the registers available within the ESP gate array and shows their visibility (ie XMI and/or Micro).

Conventions:

All address values stated in hex format.

bb - base XMI address of a given node, 2180 0000 (hex) + (80000 (hex) \* NodeID)

### 4.1 XMI REQUIRED REGISTERS

#### 4.1.1 DEVICE REGISTER (XDEV)

REGISTER : DEVICE REGISTER (XDEV)

XMI ADDRESS : bb + 0

ADAPTER MANAGER ADDRESS : 00100830

AMI ADDRESS DECODE : 0C

ACCESS : XMI READ ONLY, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```
 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
| MODULE REV | FIRMWARE | DEVICE TYPE |
+-----+-----+-----+-----+-----+-----+
```

GENERAL : THIS REGISTER IS LOADED BY THE ADAPTER MANAGER AT THE BEGINNING OF SELFTEST.

BIT DISCRPTIONS :

BITS[31:24] - This field contains the module revision level. The Adapter Manager will load this field with information retrieved from the EEPROM "KERNEL CODE" ( This is code that is always resident and never changing).

BITS[23:16] - This field contains the Adapter Manager's code revision level. The Adapter Manager will load this field with information retrieved from the EEPROM "KERNEL CODE".

BITS[15:0] - This field identifies the node type. The Adapter Manager will load this field with information retrieved from the EEPROM "KERNEL CODE".





DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY

Bit: 30                      Name: Node Reset (NRST)  
                             Access: XMI Read/Write, Adapter Manager Read Only  
                             Init Value: 0

Writing a one to this location initiates a complete power-up reset (similar to what happens in response to the assertion and deassertion of XMI DC LO L - see note below); the node performs self-test and asserts XMI BAD L until the self-test is successfully completed. Just like during power-up reset, other nodes are precluded from accessing the DEMFA from the time the bit is set to the time the DEMFA completes self-test (or the maximum self-test time is exceeded). Writing a one to this bit will cause the ESPGA to pulse a Node Reset signal to the rest of the DEMFA module for one XMI cycles

NOTE

During the time that the DEMFA is responding to node reset, the DEMFA must not access other nodes on the XMI or otherwise affect the operation of the XMI bus, but it must assert XMI BAD.

In response to a real power-up sequence (caused by XMI DC LO L), the NRST bit will be reset. Following a node reset sequence, the NRST bit will remain set, allowing the processor to recognize that it should not attempt to go through the normal boot process.

Bit: 29                      Name: Node HALT (NHALT)  
                             Access: XMI Read/Write, Adapter Manager Read Only  
                             Init Value: 0

Writing this bit to a one forces the DEMFA to go into a "quiet" state while retaining as much state as possible. When this bit is set, the ESPGA will generate an interrupt to the Adapter Manager to inform it of the Halt request. The Adapter Manager will then put the rest of the module into the "quiet" state. The Device Driver clears Node HALT.

Bit: 28                      Name: XMI BAD (XBAD)  
                             Access: Not Implemented, Read as Zero

Bit: 27                      Name: Corrected Confirmation (CC)  
                             Access: XMI Read/Write 1 to Clear  
   Adapter Manager Read Only  
                             Init Value: 0

This bit is set when the DEMFA detects a single-bit CNF error which was corrected automatically by the XCLOCK chip in the XMI Corner.

Bit: 26                      Name: XMI FAULT (XFAULT)  
                             Access: Not Implemented, Read as Zero

Bit: 25                      Name: Write Error Interrupt (WEI)  
                             Access: Not Implemented, Read as Zero

Bit: 24                      Name: Inconsistent Parity Error (IPE)  
                             Access: Not Implemented, Read as Zero

Bit: 23                      Name: Parity Error (PE)  
                             Access: XMI Read/Write 1 to Clear  
   Adapter Manager Read Only  
                             Init Value: 0

When set, indicates that the DEMFA has detected a parity error on an XMI cycle.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Bit: 22                   Name: Write Sequence Error (WSE)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a node accessing the DEMFA aborted a write transaction due to missing data cycles. Or, a write data cycle did not immediately follow a command cycle.

Bit: 21                   Name: Read/IDENT Data NoAck (RIDNAK)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a READ or IDENT data cycle transmitted by the DEMFA has received a NOACK confirmation.

Bit: 20                   Name: Write Data NoAck (WDNAK)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a WRITE data cycle transmitted by the DEMFA has received a NOACK confirmation. This bit will only be set if reattempts fail.

Bit: 19                   Name: Corrected Read Data (CRD)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that the DEMFA has received a CRDn read response.

Bit: 18                   Name: No Read Response (NRR)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a READ initiated by the DEMFA failed due to a read response timeout.

Bit: 17                   Name: Read Sequence Error (RSE)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a transaction initiated by the DEMFA failed due to a Read Sequence Error.

Bit: 16                   Name: Read Error Response (RER)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that the DEMFA received a Read Error Response.

Bit: 15                   Name: Command NoAck (CNAK)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a command cycle transmitted by the DEMFA has received a NOACK Confirmation. This can result from a reference to a Non-Existent Memory location or a command cycle parity error. This bit will only be set if reattempts fail.

Bit: 14                   Name: RESERVED  
                          Access: Read as Zero

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Bit: 13                   Name: Transaction Timeout (TTO)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read Only  
                          Init Value: 0

When set, indicates that a transaction initiated by the DEMFA failed due to a transaction timeout.

Bit: 12                   Name: Node-Specific Error Summary (NSES)  
                          Access: XMI Read Only, Adapter Manager Read/Write  
                          Init Value: 0

The Adapter Manager will set this bit when it detects a Node-Specific Error.

Bit: 11                   Name: Extended Test Fail (ETF)  
                          Access: XMI Read/Write, Adapter Manager Read/Write  
                          Init Value: 0

While set, this bit indicates that the DEMFA has not yet passed its extended test. This bit will be cleared when the node has passed its extended test. At the present time only processor nodes are allowed to implement extended test. Until changed this bit will always remain zero.

Bit: 10                   Name: Self-Test Fail (STF)  
                          Access: XMI Read/Write 1 to Clear  
                                  Adapter Manager Read/Write 1 to Clear  
                          Init Value: 1

While set, this bit indicates that the DEMFA has not yet passed its self-test. The Adapter Manager will clear this bit when the node has passed its self-test.

Bits: 9:4                 Name: Failing Commander ID (FCID)  
                          Access: XMI Read Only, Adapter Manager Read Only  
                          Init Value: Undefined

This field is used to latch the commander ID of a failing transaction. This field will be locked if any of bits <20,18:13> are asserted.

Bit: 3                    Name: Enable HexaWord Writes (EHWW)  
                          Access: XMI Read/Write, Adapter Manager Read Only  
                          Init Value: 0

This bit is used to enable/disable the transmission of hexaword writes (Hexaword Write Masked) on the DEMFA. When the EHWW bit is set, the DEMFA is permitted to generate hexaword writes. When the EHWW bit is clear, the DEMFA is restricted from generating hexaword writes.

Bit: 2                    Name: Disable XMI Timeout (DXTO)  
                          Access: XMI Read/Write, Adapter Manager Read Only  
                          Init Value: 0

This bit is used to enable/disable the reporting of all XMI timeouts by a commander. When this bit is set, the DEMFA will never encounter either TTO or RETO and is forbidden from setting XBE<13,<18. If the DEMFA has a current outstanding XMI transaction when this bit transitions from 0 to 1 (the TTO or RETO counters are counting), the given timeouts will be disabled. If the DEMFA has a current outstanding XMI transaction when this bit transitions from 1 to 0 (the TTO or RETO counters are not counting), the given timeouts will be enabled.

Bit: 1                    Name: Enable MORE Protocol (EMP)  
                          Access: XMI Read/Write, Adapter Manager Read Only  
                          Init Value: 0

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY**

This bit, when cleared, prevents the DEMFA from asserting the MORE bit on the XMI, and forces bus adapters to ignore the state of the MORE bit. When set, this bit enables the assertion of the MORE bit for the DEMFA and allows bus adapters to use the MORE bit.

Bit: 0                      Name: RESERVED  
                                 Access: Read as Zero

**4.1.3 XMI FAILING ADDRESS REGISTER (XFADR)**

REGISTER : XMI FAILING ADDRESS REGISTER (XFADR)

XMI ADDRESS : bb + 8

ADAPTER MANAGER ADDRESS : 00100838

AMI ADDRESS DECODE : 0E

ACCESS : XMI READ ONLY, ADAPTER MANAGER READ ONLY

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|FLN|                               FAILING ADDRESS                               |
+-+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

BIT DISCRIPTIONS :

BITS[31:30] - This field is used to latch the commander length of a failing transaction. This field will be locked if any of the XBER bits <20,18:13> are asserted. Otherwise these bits are loaded after each XMI command/address cycle.

BITS[29:0] - This field is used to latch the commander address of a failing transaction. This field will be locked if any of the XBER bits <20,18:13> are asserted. Otherwise these bits are loaded after each XMI command/address cycle.

**4.1.4 XMI FAILING ADDRESS EXTENTION REGISTER (XFAER)**

REGISTER : XMI FAILING ADDRESS EXTENSION REGISTER (XFAER)

XMI ADDRESS : bb + 2C

ADAPTER MANAGER ADDRESS : 0010083C

AMI ADDRESS DECODE : 0F

ACCESS : XMI READ ONLY, ADAPTER MANAGER READ ONLY

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
| FCMD |M|0| FADD EXTENSION | FAILING MASK |
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS :

- BITS[31:28] - This field is used to latch the commander command of a failing transaction. This field will be locked if any of the XBER bits <20,18:13> are asserted. Otherwise these bits will be loaded after each XMI command/address cycle.
- BIT[27] - This bit is used to latch the commander more bit of a failing transaction. This bit will be locked if any of the XBER bits <20,18:13> are asserted. Otherwise this bit will be loaded after each xmi command/address cycle.
- BIT[26] - MUST BE ZERO.
- BITS[25:16] - This field is used to latch the commander address of a failing transaction. This field will be locked if any of the XBER bits <20,18:13> are asserted. Otherwise these bits will be loaded after each xmi command/address cycle.
- BITS[15:0] - This field is used to latch the commander mask of a failing transaction. This field will be locked if any of the XBER bits <20,18:13> are asserted. Otherwise these bits will be loaded after each xmi command/address cycle.

## 4.2 DEMFA PORT REGISTERS

### 4.2.1 PORT DATA REGISTER 1 (XPD1)

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

REGISTER : PORT DATA REGISTER 1 (XPD1)

XMI ADDRESS : bb + 100

ADAPTER MANAGER ADDRESS : 00100840

AMI ADDRESS DECODE : 10

ACCESS : XMI READ/WRITE, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```
  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     DATA 1                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

BIT DISCRIPTIONS :

BITS[31:0] - This register is used by the Adapter Manager and HOST driver to exchange information. Please refer to the DEMFA Port Specification for more information.

### 4.2.2 PORT DATA REGISTER 2 (XPD2)

REGISTER : PORT DATA REGISTER 2 (XPD2)

XMI ADDRESS : bb + 104

ADAPTER MANAGER ADDRESS : 00100844

AMI ADDRESS DECODE : 11

ACCESS : XMI READ/WRITE, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```
  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     DATA 2                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

BIT DISCRIPTIONS :

BITS[31:0] - This register is used by the Adapter Manager and HOST driver to exchange information. Please refer to the DEMFA Port Specification for more information.

### 4.2.3 PORT STATUS REGISTER (XPST)

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

4

REGISTER : PORT STATUS REGISTER (XPST)

XMI ADDRESS : bb + 108

ADAPTER MANAGER ADDRESS : 00100848

AMI ADDRESS DECODE : 12

ACCESS : XMI READ ONLY, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
|                                     |E|S|W|6|F|F|F|L|PORT|
|           STATE QUALIFIER          |E|P|T|8|L|L|L|N|STATE|
|                                     |D|E|O|B|2|1|0|K|
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS :

- BITS[31:11] - This field is loaded by the Adapter Manager in order to qualify the state field. Please refer to the DEMFA Port Specification for more information.
- BIT[10] - This bit is set to a one by the Adapter Manager after a successful EEPROM update. This bit will be cleared when the HOST driver requests an EEPROM update through the EEPROM update register (EEUP). Please refer to the DEMFA Port Specification for EEPROM update information.
- BIT[9] - This bit is set when the Adapter Manager gate array discovers a SRAM parity error. When set the ESP hardware will freeze the data mover section of the gate array and generate an interrupt to the HOST.
- BIT[8] - This bit is set when the Adapter Manager gate array discovers a watch-dog timeout. When set the ESP hardware will freeze the data mover section of the gate array and generate an interrupt to the HOST.
- BIT[7] - This bit is set when the Adapter Manager gate array discovers a ESP/Adapter Manager Interface error. When set the ESP hardware will freeze the data mover section of the gate array and generate an interrupt to the HOST.
- BIT[6] - This bit has no internal function but it can be written and read by the Adapter Manager.
- BIT[5] - This bit has no internal function but it can be written and read by the Adapter Manager.
- BIT[4] - This bit has no internal function but it can be written and read by the Adapter Manager.
- BITS[3] - This bit represents the state of the FDDI link.
  - 0 = OFF
  - 1 = ON
- BITS[2:0] - The Adapter Manager loads port state information into this field. Any transition of state as represented by this field (except for into or out of reset state) will cause the ESP to generate an interrupt to the HOST. Please refer to the DEMFA Port Specification for more information.

**4.2.4 POWER UP DIAGNOSTIC REGISTER (XPUD)**

REGISTER : POWER UP DIAGNOSTIC REGISTER (XPUD)

XMI ADDRESS : bb + 10C

ADAPTER MANAGER ADDRESS : 0010084C

AMI ADDRESS DECODE : 13

ACCESS : XMI READ ONLY, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +-----+
  |                                     DATA                                     |
  +-----+
```

BIT DISCRIPTIONS :

BITS[31:0] - This register is used by the Adapter Manager to inform the HOST driver of self-test results. Please refer to the DEMFA Port Specification for more information.

**4.2.5 PORT CONTROL INITIALIZE REGISTER (XPCI)**

REGISTER : PORT CONTROL INITIALIZE REGISTER (XPCI)

XMI ADDRESS : bb + 110

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +-----+
  |                                     |I|
  |                                     |N|
  |                                     |I|
  |                                     |T|
  +-----+
```

BIT DISCRIPTIONS :

BITS[31:1] - Not implemented, bits do not exist.

BIT[0] - THE HOST driver uses this register to issue an init command to the Adapter. The command is valid when the HOST writes a one to this register. When issued, the ESP will notify the Adapter Manager of the command by asserting the "ESP TO ADAPTER MANAGER" interrupt line.

**4.2.6 PORT CONTROL SHUTDOWN REGISTER (XPCS)**



REGISTER : PORT CONTROL SHUTDOWN REGISTER (XPCS)

XMI ADDRESS : bb + 114

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|                                                     |S|
|                                                     |H|
|                                                     |U|
|                                                     |T|
+-----+

```

BIT DISCRIPTIONS :

- BITS[31:1] - Not implemented, bits do not exist.
- BIT[0] - The HOST driver uses this register to issue a shut command to the Adapter. The command is valid when the HOST writes a one to this register. When issued, the ESP will notify the Adapter Manager of the command by asserting the "ESP TO ADAPTER MANAGER" interrupt line.

**4.2.7 TRANSMIT CONTROL REGISTER (XTFL)**

REGISTER : TRANSMIT CONTROL REGISTER (XTFL)

XMI ADDRESS : bb + 118

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|                                                     |T|
|                                                     |F|
|                                                     |L|
+-----+

```

BIT DISCRIPTIONS :

- BITS[31:1] - Not implemented, bits do not exist.
- BIT[0] - The HOST driver uses this register to inform the ESP hardware that an entry(s) is available for processing on the HOST transmit ring. The action is valid when the HOST writes a one to this register.

### 4.2.8 RECEIVE CONTROL REGISTER (XRFL)

REGISTER : RECEIVE CONTROL REGISTER (XRFL)

XMI ADDRESS : bb + 11C

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|                                                     |R|
|                                                     |F|
|                                                     |L|
+-----+
```

BIT DISCRIPTIONS :

BITS[31:1] - Not implemented, bits do not exist.

BIT[0] - The HOST driver uses this register to inform the ESP hardware that an entry(s) is available for processing on the HOST receive ring. The action is valid when the HOST writes a one to this register.

### 4.2.9 COMMAND CONTROL REGISTER (XCFL)

REGISTER : COMMAND CONTROL REGISTER (XCFL)

XMI ADDRESS : bb + 120

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|                                                     |C|
|                                                     |F|
|                                                     |L|
+-----+
```

BIT DISCRIPTIONS :

BITS[31:1] - Not implemented, bits do not exist.

BIT[0] - The HOST driver uses this register to inform the Adapter Manager that an entry(s) is available for processing on the HOST command ring. The action is valid when the HOST writes a one to this register. When issued, the ESP will notify the Adapter Manager of the command by asserting the "ESP TO ADAPTER MANAGER" interrupt line.

#### 4.2.10 UNSOLICITED CONTROL REGISTER (XUFL)

REGISTER : UNSOLICITED CONTROL REGISTER (XUFL)

XMI ADDRESS : bb + 124

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|                                               |U|
|                                               |F|
|                                               |L|
+-----+
```

BIT DISCRIPTIONS :

- BITS[31:1] - Not implemented, bits do not exist.
- BIT[0] - The HOST driver uses this register to inform the Adapter Manager that an entry(s) is available for processing on the HOST unsolicited ring. The action is valid when the HOST writes a one to this register. When issued, the ESP will notify the Adapter Manager of the command by asserting the "ESP TO ADAPTER MANAGER" interrupt line.

#### 4.2.11 TRANSMIT HOST HIBERNATION LOW REGISTER (THIBL)

REGISTER : TRANSMIT HOST HIBERNATION LOW REGISTER (THIBL)

XMI ADDRESS : bb + 128

ADAPTER MANAGER ADDRESS : 00100850

AMI ADDRESS DECODE : 14

ACCESS : XMI READ/WRITE, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|          HOST TRANSMIT HIBERNATION REGISTER          |
+-----+
```

BIT DISCRIPTIONS :

- BITS[31:0] - This register is loaded by the HOST driver after it has discovered that no more transmit entries are available to process. This register contains the lower 32 bits of the address that points to the last successfully processed transmit buffer discriptor. The Adapter Manager is given access to this register for testing purposes.

#### 4.2.12 TRANSMIT HOST HIBERNATION HIGH REGISTER (THIBH)

REGISTER : TRANSMIT HOST HIBERNATION HIGH REGISTER (THIBH)

XMI ADDRESS : bb + 12C

ADAPTER MANAGER ADDRESS : 00100854

AMI ADDRESS DECODE : 15

ACCESS : XMI READ/WRITE, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
    +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    |                                     | ADDRESS |
    +-----+-----+-----+-----+-----+-----+-----+-----+
    
```

BIT DISCRIPTIONS :

BITS[31:8] - Must be zero.

BITS[7:0] - This register is loaded by the HOST after it has discovered that no more transmit entries are available to process. This register contains the upper 8 bits of the address that points to the last successfully processed transmit buffer discriptor. The Adapter Manager is given access to this register for testing purposes.

#### 4.2.13 RECEIVE HOST HIBERNATION LOW REGISTER (RHIBL)

REGISTER : RECEIVE HOST HIBERNATION LOW REGISTER (RHIBL)

XMI ADDRESS : bb + 130

ADAPTER MANAGER ADDRESS : 00100858

AMI ADDRESS DECODE : 16

ACCESS : XMI READ/WRITE, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
    +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    |                                     | HOST RECEIVE HIBERNATION REGISTER |
    +-----+-----+-----+-----+-----+-----+-----+-----+
    
```

BIT DISCRIPTIONS :

BITS[31:0] - This register is loaded by the HOST after it has discovered that no more receive entries are available to process. This register contains the lower 32 bits of the address that points to the last successfully processed receive buffer discriptor. The Adapter Manager is given access to this register for testing purposes.

#### 4.2.14 RECEIVE HOST HIBERNATION HIGH REGISTER (RHIBH)

REGISTER : RECEIVE HOST HIBERNATION HIGH REGISTER (RHIBH)

XMI ADDRESS : bb + 134

ADAPTER MANAGER ADDRESS : 0010085C

AMI ADDRESS DECODE : 17

ACCESS : XMI READ/WRITE, ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     | ADDRESS |
+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRPTIONS :

BITS[31:8] - Must be zero.

BITS[7:0] - This register is loaded by the HOST after it has discovered that no more receive entries are available to process. This register contains the upper 8 bits of the address that points to the last successfully processed receive buffer descriptor. The Adapter Manager is given access to this register for testing purposes.

**4.2.15 EEPROM UPDATE REGISTER (EUP)**

REGISTER : EEPROM UPDATE REGISTER (EUP)

XMI ADDRESS : bb + 138

ADAPTER MANAGER ADDRESS : NONE

AMI ADDRESS DECODE : NONE

ACCESS : XMI WRITE ONLY

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |E|
|                                     |E|
|                                     |U|
|                                     |P|
+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRPTIONS :

BITS[31:1] - Not implemented, bits do not exist.

BIT[0] - The HOST driver uses this register to issue an EEPROM update to the Adapter. The command is valid when the HOST writes a one to this register. Please refer to the DEMFA Port Specification for EEPROM update information.

**4.2.16 INTERRUPT VECTOR REGISTER (ITVR)**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY**

REGISTER : INTERRUPT VECTOR REGISTER (ITVR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100860

AMI ADDRESS DECODE : 18

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|           | IPL |           INTR DESTINATION           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS :

BITS[31:20] - MUST BE ZERO.

BITS[19:16] - This field is loaded by the Adapter Manager with one of the four device interrupt levels. This information is retrieved from the DEMFA port data block.

BITS[15:0] - This field is loaded by the Adapter Manager with the destination mask of which CPU nodes the interrupt can be serviced by. Please refer to the XMI Specification for more information on interrupts.

**4.2.17 INTERRUPT DESTINATION VECTOR REGISTER (IDVR)**

REGISTER : INTERRUPT DESTINATION VECTOR REGISTER (IDVR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100864

AMI ADDRESS DECODE : 19

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|           |           DESTINATION VECTOR           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS :

BITS[31:16] - Must be zero.

BITS[15:4] - This field is loaded by the Adapter Manager with the vector information taken from the port data block.

BITS[3:0] - Must be zero.

**4.3 ESP CONTROL AND STATUS REGISTERS**

### 4.3.1 ESP ERROR REGISTER (ESPERR)

REGISTER : ESP ERROR REGISTER (ESPERR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100888

AMI ADDRESS DECODE : 22

ACCESS : SEE BIT DISCRPTIONS.

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
    +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    |                                                                 |S|P|I|X|M|6|P|X|P|T|R|T|R| |
    |                                                                 |M|G|P|M|O|8|B|B|B|F|F|O|O| |
    |                                                                 |E|E|L|I|W|P|P|C|C|T|T|W|W| |
    |                                                                 | | |E|E|N|E|E|E|E|E|N|N| |
    +-----+-----+-----+-----+-----+-----+-----+-----+
    
```

BIT DISCRPTIONS :

Bit: 31:14 Not Implemented, read as zero.

Bit: 13 Name: State Machine Error (SME).  
 Access: Adapter Manager R/W 1 to clear.

This bit is Set if the ESP detects any of it's internal State Machines entering an illegal state.

Bit: 12 Name: Parity Generation Error (PGE).  
 Access: Adapter Manager Read/Write 1 to Clear.

This bit is Set if the ESP interface detects a Parity Generation Error when converting XMI longword parity to internal byte parity.

Bit: 11 Name: IPL Error (IPLE)  
 Access: Adapter Manager Read/Write 1 to Clear

This bit is set if the ESP receives an Ident command with an IPL that doesn't match the interrupt IPL that was sent by the DEMFA. This bit denotes a fatal condition.

Bit: 10 Name: XMI Error (XMIE)  
 Access: Adapter Manager Read Only  
 Init Value: 0

This bit is the logical-OR of XBE Register bits <27,22:20,18:15,13>. This bit denotes a fatal condition.

Bit: 9 Not Implemented. Read as 0.

Bit: 8 Name: ESP/AMI Parity Error (68PE)  
 Access: Adapter Manager Read/Write 1 to Clear  
 Init Value: 0

This bit is set, when the ESP detects a parity error on it's interface bus with the AMI gate array. This bit denotes a fatal condition.

Bit: 7 Name: ESP/PMC Parity Error (PBPE)  
 Access: Adapter Manager Read/Write 1 to Clear  
 Init Value: 0

This bit is set, when the ESP detects a parity error on it's interface bus with the PMC gate array. This bit denotes a fatal condition.

Bit: 6 Name: XMI Byte Count Error (XBCE)  
 Access: Adapter Manager Read/Write 1 to Clear  
 Init Value: 0

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

This bit is set, when an illegal byte count is detected by the ESP byte count logic. This bit denotes a fatal condition.

Bit: 5                    Name: PBI Byte Count Error (PBCE)  
                         Access: Adapter Manager Read/Write 1 to Clear  
                         Init Value: 0

This bit is set, when an illegal byte count is detected by the ESP byte count logic. This bit denotes a fatal condition.

Bit: 4                    Name: Transmit Format Error (TFTE)  
                         Access: Adapter Manager Read/Write 1 to Clear  
                         Init Value: 0

This bit is set, when an illegal Transmit Entry format is detected by the ESP transmit logic. This bit denotes a fatal condition.

Bit: 3                    Name: Receive Format Error (RFTE)  
                         Access: Adapter Manager Read/Write 1 to Clear  
                         Init Value: 0

This bit is set, when an illegal Receive Buffer format is detected by the ESP receive logic. This bit denotes a fatal condition.

Bit: 2                    Name: Transmit Own Error (TOWN)  
                         Access: Adapter Manager Read/Write 1 to Clear  
                         Init Value: 0

This bit is set, when the ESP transmit logic detects that it does not own a page that it expects to own. This bit denotes a fatal condition.

Bit: 1                    Name: Receive Own Error (ROWN)  
                         Access: Adapter Manager Read/Write 1 to Clear  
                         Init Value: 0

This bit is set, when the ESP receive logic detects that it does not own a page that it expects to own. This bit denotes a fatal condition.

Bit: 0                    Not Implemented, read as zero.

NOTE:

All bit definitions with fatal condition statements will generate an interrupt to the 68020.

### 4.3.2 ADAPTER MANAGER INTERRUPT REGISTER (AMIR)



REGISTER : ADAPTER MANAGER INTERRUPT REGISTER (AMIR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100868

AMI ADDRESS DECODE : 1A

ACCESS : ADAPTER MANAGER READ/WRITE 1 TO CLEAR

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+
|                                     |N|E|U|C|P|P|H|P|E|
|                                     |B|E|F|F|C|C|L|M|S|
|                                     |A|U|L|L|S|I|T|C|P|
+-----+

```

GENERAL : This register is read by the Adapter Manager to determine the source(s) of the "ESP TO ADAPTER MANAGER" interrupt(s).

BIT DISCRIPTIONS :

- BITS[31:9] - Not implemented, read as zero.
- BIT[8] - This bit is set when the ESP detects that there no longer exists any free receive buffers.
- BIT[7] - This bit is set when the HOST driver issues an EEPROM update through the EEPROM update register.
- BIT[6] - This bit is set when the HOST driver writes the unsolicited control register.
- BIT[5] - This bit is set when the HOST driver writes the command control register.
- BIT[4] - This bit is set when the HOST driver issues a shut command through the port control shutdown register.
- BIT[3] - This bit is set when the HOST driver issues an init command through the port control initialize register.
- BIT[2] - This bit is set when the HOST driver writes the XBE register halt bit (29).
- BIT[1] - This bit is set when the PMC gate array wants to interrupt the Adapter Manager. Please refer to the PMC Specification for more information.
- BIT[0] - This bit is set when the ESP error register has one or more of the following bits set: <10,8:1>.

**4.3.3 ESP MISCELLANEOUS REGISTER (ESPMR)**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

REGISTER : ESP MISCELLANEOUS REGISTER (ESPMR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 0010086C

AMI ADDRESS DECODE : 1B

ACCESS : SEE BIT DISCRIPTIONS

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
|                                     |U|F|F|C|H|A|A|F|F|I|F|D|B|R|L|S|E|C|
|                                     |P|B|B|H|P|X|P|B|D|D|T|R|A|S|B|L|H|U|
|                                     |D|3|2|E|R|E|E|1|M|M|O|Y|D|T|D|B|I|I|
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS :

- Bit: 31:18 Not Implemented, read as zero.
- Bit: 17 Name: Update Enable (UPD).  
Access: AM Read.  
Initialization Value: Backplane Dependent.  
The Adapter Manager must make sure this bit is SET before an update of the EEPROM is performed.
- Bit: 16 Name: Force Bad Parity 3 (FB3)  
Access: Adapter Manager Read/Write 1 to Set/  
Write 1 to Clear  
Init Value: 0

The Adapter Manager can set this bit to test the parity detection circuitry inside the ESP gate array on the XCI Bus. When set to a one, this bit will cause the parity bits entering the ESP on that bus to be flipped, such that a data transaction will cause a parity error.

- Bit: 15 Name: Force Bad Parity 2 (FB2)  
Access: Adapter Manager Read/Write 1 to Set/  
Write 1 to Clear  
Init Value: 0

The Adapter Manager can set this bit to test the parity detection circuitry inside the ESP gate array on the "ESP to PMC" Bus. When set to a one, this bit will cause the parity bits entering the ESP on that bus to be flipped, such that a data transaction will cause a parity error.

- Bit: 14 Name: Check Host Ring On Completed Transmission (CHE)  
Access: Adapter Manager Read/Write 1 to Set/  
Write 1 to Clear  
Init Value: 0

When cleared, the ESP hardware that reads the Host Transmit Ring will make an additional check for new entries at the completion of an existing transmitted packet, if it is currently in a "sleep" state. This is provided to ensure that the ESP date mover can always be updated with current information.

- Bit: 13 Name: Adapter Manager XMI Priority (HPR)  
Access: Adapter Manager Read/Write 1 to Set/  
Write 1 to Clear  
Init Value: 0

The Adapter Manager will set this bit to a one to enable the ESP hardware to perform an Adapter Manager XMI transaction at the highest priority level with respect to the other XMI processes that exist in the ESP. While zero, the Adapter Manager has the lowest XMI priority.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Bit: 12                   Name: Adapter Manager XMI Transaction Enable (AXE)  
                          Access: Adapter Manager Read/Write 1 to Set/  
  Write 1 to Clear  
                          Init Value: 0

The Adapter Manager will set this bit to a one to enable the ESP hardware to perform an Adapter Manager XMI transaction which occurs as a function of the Adapter Manager writing the Adapter Manager XMI Low Address Register or writing the Adapter Manager XMI High Address Register's bit 25 (Discard).

Bit: 11                   Name: Adapter Manager PBI Transaction Enable (APE)  
                          Access: Adapter Manager Read/Write 1 to Set/  
  Write 1 to Clear  
                          Init Value: 0

The Adapter Manager will set this bit to a one to enable the ESP hardware to perform an Adapter Manager PBI transaction which occurs as a function of the Adapter Manager writing the Adapter Manager Packet Buffer Address Register.

Bit: 10                   Name: Force Bad Parity 1 (FB1)  
                          Access: Adapter Manager Read/Write 1 to Set/  
  Write 1 to Clear  
                          Init Value: 0

The Adapter Manager can set this bit to test the parity detection circuitry inside the ESP gate array on the Adapter Manager Bus. When set to a one, this bit will cause the parity bits entering the ESP on that bus to be flipped, such that a data transaction will cause a parity error. The ESP allows writing of registers when parity checking is even and parity errors are happening. This lets you reset the parity checking to odd in order to stop parity errors.

Bit: 9                    Name: Freeze Data Mover (FDM)  
                          Access: Adapter Manager Read as 0/Write 1 to Set  
                          Init Value: 0

The Adapter Manager can use this bit to prevent the Data Mover from generating XMI traffic. During exception errors (XPST bits 9:7 being set), this bit will automatically be set. Once frozen, the ESP must receive either a complete initialization or an Init Data Mover to become revived. When read, this bit is 0.

Bit: 8                    Name: Init Data Mover (IDM)  
                          Access: Adapter Manager Read/Write 1 to Pulse  
                          Init Value: 0

When written with a one by the Adapter Manager, a pulse will be generated which causes the Data Mover state machines and data queues to initialize to a ready state. This bit resets the rest of the module. Note that XPST, XDEV, and STF bits are not cleared by this function. This bit also resets the fatal error code bits in the XPST register.

Bit: 7                    Name: Force Time Out (FTO)  
                          Access: Adapter Manager Read/Write 1 to Set/  
  Write 1 to Clear  
                          Init Value: 0

When set, the XMI transaction timeout counter will timeout in a fraction of the time normally used. This bit is strictly used for testing.

Bit: 6                    Name: Disable XMI Retry (DRY)  
                          Access: Adapter Manager Read/Write 1 to Set/  
  Write 1 to Clear  
                          Init Value: 0



REGISTER : HOST TRANSMIT RING BASE LOW ADDRESS REGISTER (HTRBLAR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100870

AMI ADDRESS DECODE : 1C

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     BASE ADDRESS LOW                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the low part, bits 31-0, of the  
 HOST transmit ring's base address.

### 4.3.5 HOST TRANSMIT RING BASE HIGH ADDRESS AND SIZE REGISTER (HTRBHARSR)

REGISTER : HOST TRANSMIT RING BASE HIGH ADDRESS AND RING SIZE REGISTER (HTRBHARSR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100874

AMI ADDRESS DECODE : 1D

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     |BASE ADDR HIGH |          RING SIZE          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:20] - Must be zero.  
 BITS [19:12] - These bits contain the high part, bits 39-32, of the  
 HOST transmit ring's base address.  
 BITS [11:0] - These bits contain the HOST transmit ring size, in bytes.

### 4.3.6 HOST RECEIVE RING BASE LOW ADDRESS REGISTER (HRRBLAR)

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY**

REGISTER : HOST RECEIVE RING BASE LOW ADDRESS REGISTER (HRRBLAR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100878

AMI ADDRESS DECODE : 1E

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     BASE ADDRESS LOW                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the low part, bits 31-0, of the  
HOST receive ring's base address.

**4.3.7 HOST RECEIVE RING BASE HIGH ADDRESS AND SIZE REGISTER (HRRBHARSR)**

REGISTER : HOST RECEIVE RING BASE HIGH ADDRESS, RING SIZE AND  
ENTRY SIZE REGISTER (HRRBHARSR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 0010087C

AMI ADDRESS DECODE : 1F

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|BASE ADDR HIGH |          ENTRY SIZE          |          RING SIZE          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

BIT DISCRIPTIONS -

BITS [31:24] - These bits contain the high part, bits 39-32, of the  
HOST receive ring's base address.

BITS [23:12] - These bits contain the HOST receive ring's entry size,  
in bytes.

BITS [11:0] - These bits contain the HOST receive ring size, in bytes.

**4.3.8 PMC TRANSMIT RING BASE ADDRESS REGISTER (PMCTRBAR)**

REGISTER : PMC TRANSMIT RING BASE ADDRESS AND RING SIZE REGISTER (PMCTRBAR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100880

AMI ADDRESS DECODE : 20

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   |           RING SIZE           |           BASE ADDRESS           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

- BITS [31:30] - Must be zero.
- BITS [29:15] - These bits contain the PMC transmit ring size, in pages.
- BITS [14:0] - These bits contain the PMC transmit base address page frame number.

**4.3.9 PMC RECEIVE RING BASE ADDRESS REGISTER (PMCRBAR)**

REGISTER : PMC RECEIVE RING BASE ADDRESS AND RING SIZE REGISTER (PMCRBAR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100884

AMI ADDRESS DECODE : 21

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   |           RING SIZE           |           BASE ADDRESS           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

- BITS [31:30] - Must be zero.
- BITS [29:15] - These bits contain the PMC receive ring size, in pages.
- BITS [14:0] - These bits contain the PMC receive base address page frame number.

**4.3.10 ADAPTER MANAGER PACKET BUFFER ADDRESS REGISTER (AMPBAR)**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

REGISTER : ADAPTER MANAGER PACKET BUFFER ADDRESS REGISTER (AMPBAR)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100804

AMI ADDRESS DECODE : 01

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|SIZE |A|A|C|B|S|           ADDRESS/SPECIAL           |R|T|
|      |C|M|O|D|F|                                           |W|R|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

BIT DISCRIPTIONS -

BITS [31:29] - These bits contain the transaction length, in Lws minus one, for the Adapter Manager's packet buffer operation.

GENERAL LENGTH CODES

- ```

-----
000 - one lw buffer read/write
      buffer discriptor read/write
      clear own cycle
001 - two lw buffer read/write
010 - three lw buffer read/write
011 - four lw buffer read/write

```

- BIT [28] - This bit can be set if the Adapter Manager wants to automatically clear the ownership of a buffer after a buffer discriptor write operation.
- BIT [27] - This bit must be set to a one by the Adapter Manager for all packet buffer operations.
- BIT [26] - This bit, when set, signifies a clear ownership cycle.
- BIT [25] - This bit must be zero, for buffer read/write operations and one, for buffer discriptor and clear ownership operations.
- BIT [24] - This bit, when clear, signifies that bits [23:2] contain valid packet buffer address information. This bit, when set, signifies that bits [23:2] hold a special function as specified in the PMC Specification.
- BIT [23:2] - Depending on the value of bit [24], this field may contain address information or special information.
- BIT [1] - This bit, when set, indicates that the Adapter Manager is attempting to perform a read/type operation. This bit, when clear, indicates that the Adapter Manager is attempting to perform a write/type operation.
- BIT [0] - This bit, when set, indicates that the Adapter Manager is attempting to perform a transection associated with a transmit function. This bit, when clear, indicates that the Adapter Manager is attempting to perform a transaction associated with a receive function.



### 4.3.11 ADAPTER MANAGER PACKET BUFFER DATA 0 REGISTER (AMPBD0R)

REGISTER : ADAPTER MANAGER PACKET BUFFER DATA 0 REGISTER (AMPBD0R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100810

AMI ADDRESS DECODE : 04

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     READ/WRITE DATA                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager packet buffer transaction. On a write into the packet buffer, this register will hold write data previously loaded by the Adapter Manager. On a read from the packet buffer, this register will hold read data for the Adapter Manager.

### 4.3.12 ADAPTER MANAGER PACKET BUFFER DATA 1 REGISTER (AMPBD1R)

REGISTER : ADAPTER MANAGER PACKET BUFFER DATA 1 REGISTER (AMPBD1R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100814

AMI ADDRESS DECODE : 05

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     READ/WRITE DATA                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager packet buffer transaction. On a write into the packet buffer, this register will hold write data previously loaded by the Adapter Manager. On a read from the packet buffer, this register will hold read data for the Adapter Manager.

### 4.3.13 ADAPTER MANAGER PACKET BUFFER DATA 2 REGISTER (AMPBD2R)

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

REGISTER : ADAPTER MANAGER PACKET BUFFER DATA 2 REGISTER ( AMPBD2R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100818

AMI ADDRESS DECODE : 06

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
|                                     READ/WRITE DATA                                     |
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager packet buffer transaction. On a write into the packet buffer, this register will hold write data previously loaded by the Adapter Manager. On a read from the packet buffer, this register will hold read data for the Adapter Manager.

**4.3.14 ADAPTER MANAGER PACKET BUFFER DATA 3 REGISTER (AMPBD3R)**

REGISTER : ADAPTER MANAGER PACKET BUFFER DATA 3 REGISTER (AMPBD3R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 0010081C

AMI ADDRESS DECODE : 07

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
|                                     READ/WRITE DATA                                     |
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager packet buffer transaction. On a write into the packet buffer, this register will hold write data previously loaded by the Adapter Manager. On a read from the packet buffer, this register will hold read data for the Adapter Manager.

**4.3.15 ADAPTER MANAGER XMI LOW ADDRESS REGISTER (AMXMIL)**

REGISTER : ADAPTER MANAGER XMI LOW ADDRESS REGISTER (AMXMIL)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100808

AMI ADDRESS DECODE : 02

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ADDRESS LOW                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the low part, bits 31 - 0, of the address used in the XMI transaction.

**4.3.16 ADAPTER MANAGER XMI HIGH ADDRESS REGISTER (AMXMIH)**

REGISTER : ADAPTER MANAGER XMI HIGH ADDRESS REGISTER

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 0010080C

AMI ADDRESS DECODE : 03

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|           |D|C|           MASK           | ADDRESS HIGH |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

- BITS [31:26] - Must be zero.
- BIT [25] - When set, the ESP hardware will initiate an XMI discard transaction.
- BIT [24] - When set, this bit indicates an XMI read transaction. When clear, this bit indicates an XMI write transaction.
- BITS [23:8] - These bits are the mask field used during write-type operations to validate the bytes used in the transaction.
- BITS [7:0] - These bits contain the high part, bits 40 - 32, of the address used in the XMI transaction.

**4.3.17 ADAPTER MANAGER XMI DATA 0 REGISTER (AMXMID0R)**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY**

REGISTER : ADAPTER MANAGER XMI DATA 0 REGISTER (AMXMID0R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100820

AMI ADDRESS DECODE : 08

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     READ/WRITE DATA                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager XMI transaction. On a write into the HOST memory, this register will hold write data previously loaded by the Adapter Manager. On a read from the HOST memory, this register will hold read data for the Adapter Manager.

**4.3.18 ADAPTER MANAGER XMI DATA 1 REGISTER (AMXMID1R)**

REGISTER : ADAPTER MANAGER XMI DATA 1 REGISTER (AMXMID1R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100824

AMI ADDRESS DECODE : 09

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     READ/WRITE DATA                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager XMI transaction. On a write into the HOST memory, this register will hold write data previously loaded by the Adapter Manager. On a read from the HOST memory, this register will hold read data for the Adapter Manager.

**4.3.19 ADAPTER MANAGER XMI DATA 2 REGISTER (AMXMID2R)**

REGISTER : ADAPTER MANAGER XMI DATA 2 REGISTER (AMXMID2R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 00100828

AMI ADDRESS DECODE : 0A

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
|                               READ/WRITE DATA                               |
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager XMI transaction. On a write into the HOST memory, this register will hold write data previously loaded by the Adapter Manager. On a read from the HOST memory, this register will hold read data for the Adapter Manager.

**4.3.20 ADAPTER MANAGER XMI DATA 3 REGISTER (AMXMID3R)**

REGISTER : ADAPTER MANAGER XMI DATA 3 REGISTER (AMXMID3R)

XMI ADDRESS : NONE

ADAPTER MANAGER ADDRESS : 0010082C

AMI ADDRESS DECODE : 0B

ACCESS : ADAPTER MANAGER READ/WRITE

INITIAL VALUE : 0

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+
|                               READ/WRITE DATA                               |
+-----+-----+-----+-----+-----+-----+

```

BIT DISCRIPTIONS -

BITS [31:0] - These bits contain the data used during an Adapter Manager XMI transaction. On a write into the HOST memory, this register will hold write data previously loaded by the Adapter Manager. On a read from the HOST memory, this register will hold read data for the Adapter Manager.

# **CHAPTER 5**

## **AMI REGISTERS**

This chapter describes all the registers available within the AMI gate array.

**Conventions:**

All address values stated in hex format. Byte, word and longword addressing conforms to the Motorola big-endian Protocol ( Byte 0 is always the most significant D[31:24] byte ).

Figure 17: MEMORY MAP

\$FFFF FFFF

|             |                               |        |
|-------------|-------------------------------|--------|
|             | NOT DECODED                   | < 16GB |
| \$0400 0000 | BUFFER DESCRIPTOR SPACE       | 16MB   |
| \$0300 0000 | PAGE - TABLE SPACE            | 16MB   |
| \$0200 0000 | PACKET MEMORY SPACE           | 16MB   |
| \$0100 0000 | NOT DECODED                   | 13MB   |
| \$0030 0000 | RESERVED FOR SRAM EXPANSION   | 768kB  |
| \$0024 0000 | SRAM                          | 256KB  |
| \$0020 0000 | RESERVED                      | < 1MB  |
| \$0010 2000 | PMC CSR SPACE                 | 1KB    |
| \$0010 1C00 | SELF TEST SPACE               | 1KB    |
| \$0010 1800 | INTERRUPT CONTROLLER          | 1KB    |
| \$0010 1400 | 68S CSRs                      | 1KB    |
| \$0010 1000 | COUNTER - TIMERS              | 1KB    |
| \$0010 0C00 | ESP BUS INTERFACE             | 1KB    |
| \$0010 0800 | FDDI CSR BUS INTERFACE        | 1KB    |
| \$0010 0400 | DPA ROM                       | 1KB    |
| \$0010 0000 | RESERVED FOR EEPROM EXPANSION | 768KB  |
| \$0004 0000 | EEPROM                        | 256KB  |
| \$0000 0000 |                               |        |

## Dynamic Bus Sizing

The MC68020 dynamically interprets the port size of the addressed register within the AMI Gate Array during each cycle, allowing transfers to or from 8-, 16-, and 32-bit ports. During a transfer cycle, the slave device signals its port size (byte, word, or longword) and indicates completion of the bus cycle to the processor through the use of the DSACKL[1:0] signal lines.

For example if the 68020 is executing an instruction that reads a longword from a longword aligned address, it attempts to read 32 bits during the first bus cycle. If the port responds that it is 16 bits wide, the 68020 latches 16 bits of valid data ( D[31:16] ) and runs another bus cycle to obtain the other 16 bits ( again on D[31:16] ). For 16-bit word reads, the AMI drives D[15:0] with all zeros.

## 5.1 ESP INTERFACE STATUS REGISTER (ESPIFSR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0A00

=====

Access: RO ( Read Only )

=====

Initialized as: All bits zero.

=====

```

    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|   |H|P|
|   |B|B|
+-----+-----+-----+-----+-----+-----+-----+-----+
    
```

Bit Descriptions:

=====

BIT<31:02>: NIO ( Not Implemented, Read as zero )

BIT<01>: Host/XMI Busy; 1 = Busy, 0 = Not Busy

This status bit when set informs the 68020 that an ESP to Host Transaction is in progress. This bit will be set by the ESP G.A. when the 68020 writes the AMXMIH Register. While this bit is set, the 68020 must not initiate a new transaction by writing the AMXMIH Register again. Once the bit is cleared, a new transaction can be initiated.

BIT<00>: PMC Busy; 1 = Busy, 0 = Not Busy

This status bit when set informs the 68020 that an ESP to PMC Transaction is in progress. This bit will be set by the ESP G.A. when the 68020 writes the AMPBAR Register. While this bit is set, the 68020 must not initiate a new transaction by writing the AMPBAR Register again. Once the bit is cleared, a new transaction can be initiated.



## 5.2 TIMERS AND COUNTERS

### 5.2.1 AMI WATCHDOG TIMER (WDTIMR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C00  
 =====

Access: R/W ( Read / Write )  
 =====

Initialized as: All bits zero.  
 =====

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +-+
  |                               Counter Preset Value                               |
  +-+
  
```

Bit Descriptions:  
 =====

BIT<31:00>: Data to be loaded into the Watchdog Counter as a preset value. The counter used for the Watchdog Timer is 32-Bits wide. Bit<31> is the Most Significant Counter Bit and Bit<0> is the Least Significant Counter bit. Each binary bit weight is multiplied times the 64nsec clock period to calculate the counter expiration time. The counter will decrement from this preset value and interrupt the 68020 when it reaches \$0000 0000.

A readback of this register will provide the latest counter value at the time of the 68020 read cycle.

This preset value is reloaded into the counter and a new count down sequence will start whenever the Watchdog Timer is cleared by firmware.

| The maximum watchdog time is 274 seconds.

### 5.2.2 AMI RTOS TIMER (RTOSTIMR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C04  
 =====

Access: R/W ( Read / Write )  
 =====

Initialized as: All bits zero.  
 =====

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +-+
  |0|0|0|0|0|0|0|0|0|0|0|0|                               Counter Preset Value                               |
  +-+
  
```

Bit Descriptions:  
 =====

BIT<31:24>: MBZ ( Must Be Zero )

BIT<23:00>: Data to be loaded into the RTOS Counter as a preset value. The counter used for the RTOS Timer is 24-Bits wide allowing an elapsed time of up to 1 second before initiating a 68020 RTOS Timer Interrupt.

Bit<23> is the Most Significant Counter Bit and Bit<0> is the Least Significant Counter Bit. Each binary bit weight is multiplied times the 64nsec clock period to calculate the counter expiration time. The counter will decrement from this preset value on each clock cycle and generate an interrupt to the 68020 when it reaches \$0000 0000. Once enabled, this Counter will free run by counting down, reloading the initial counter value and repeating the count down sequence.

A readback of this register will provide the latest counter value at the time of the 68020 read cycle.

### 5.2.3 COUNTER/TIMER CSR (CTCSR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C08  
 =====

Access: R/W ( Read / Write )  
 =====

Initialized as: All bits zero.  
 =====

```

3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
	B	R	W	B	R	W
	E	T	D	E	T	D
	C	C	S	T	T	T
	T	T	E	E	E	
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    
```

Bit Descriptions:  
 =====

BIT<31:06>: NIO ( Not implemented, Read as zero )

BIT<05>: Bus Error Counter Self Test Mode; 1 = Enabled, 0 = Disabled  
 This bit is for Bus Error Counter Self Test. It allows one to test the operation of the Bus Error Timer by chaining the counter stages together to minimize test time. When this bit is at a logic "1", the Bus Error Counters will be chained together. When set to the default logic "0", the counters will be set for normal operation. This bit is Read/Write access.

BIT<04>: RTOS Counter Self Test Mode; 1 = Enabled, 0 = Disabled  
 This bit is for RTOS Counter Self Test. It allows one to test the operation of the RTOS Timer by chaining the counter stages together to minimize test time. When this bit is at a logic "1", the RTOS Counters will be chained together. When set to the default logic "0", the counters will be set for normal operation. This bit is Read/Write access.

BIT<03>: Watchdog Timer Self Test Mode; 1 = Enabled, 0 = Disabled  
 This bit is for Self Test Code to test the operation of the Watchdog Timer by chaining the counter stages together to minimize test time. When this bit is at a logic "1", the Watchdog Counters will be chained together. When set to the default logic "0", the counters will be set for normal operation. This bit is Read/Write access.

BIT<02>: Bus Error Timer Enable; 1 = Enable, 0 = Disable.  
 When set to logic "1" this bit will enable the Bus Error Counter to be operational. The value of BET Register should be loaded prior to setting this bit. The default value of this bit at RESET is zero for counter disabled. This bit is Read/Write Access.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Once enabled, the counter will start counting at the beginning of a 68020 Bus Cycle and be reloaded at the end of a 68020 Bus Cycle. When set to a logic "0" this bit will disable all Bus Error Counter operation.

BIT<01>: RTOS Timer Enable; 1 = Enable, 0 = Disable.  
 When set to logic "1" this bit will enable the RTOS Counter to be operational. The value of RTOSTIMR Register should be loaded prior to setting this bit. The default value of this bit at RESET is zero. This bit is Read/Write Access.

The counter will start counting once enabled and free run. When set to a logic "0", this bit will disable RTOS Counter operation.

BIT<00>: Watchdog Timer Enable; 1 = Enable, 0 = Disable.  
 When set to logic "1" this bit will enable the Watchdog Counter to be operational. The value of WDTIMR Register should be loaded prior to setting this bit. The default value of this bit at RESET is zero for counter disabled. This bit is Read/Write Access.

Once enabled, the counter will start counting down. In normal operation, firmware will clear the counter before it reaches a count of \$0000 0000 and issues a Watchdog Timer Interrupt. The Watchdog Timer is cleared and reloaded with a write to WDTCLR Register.

**5.2.4 BUS ERROR TIMER (BET)**

Adapter Manager ( 68020 ) Physical Address: \$0010 0C10

Access: R/W ( Read / Write )

Initialized as: \$0000 00FF ( approx 16usec. )

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +-----+-----+-----+-----+-----+-----+-----+-----+
  |0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
                                     Counter Preset
  +-----+-----+-----+-----+-----+-----+-----+-----+
  
```

Bit Descriptions:

BIT<31:16>: MBZ ( Must Be Zero )

BIT<15:00>: Data to be loaded into the Bus Error Counter as a preset value. The counter used for the Bus Error Timer is 16-Bits wide allowing an elapsed time of up to 4 msec. before initiating a 68020 Bus Error Cycle.

Bit<15> is the Most Significant Counter Bit and Bit<0> is the Least Significant Counter Bit. Each binary bit weight is multiplied times the 64nsec clock period to calculate the counter expiration time. The counter will decrement from this preset value on each clock cycle and generate a Bus Error to the 68020 when it reaches \$0000 0000. The Bus Error Counter will be enabled and start counting whenever a new 68020 Bus Cycle starts. The counter will be stopped and reloaded when DSACKL[1:0] is asserted. The counter is reloaded to the value written in the register above at the normal completion of a 68020 Bus Cycle.

A readback of this register will provide the latest counter value at the time of the 68020 read cycle. A default value for the register other than zero is provided to assist Self Test Code.

### 5.3 AMI GENERAL PURPOSE CSR (GPCSR)

Adapter Manager ( 68020 ) Physical Address: \$0010 1000  
 =====

Access: SC ( Special Case - see bit descriptions )  
 =====

Initialized as: All bits zero.  
 =====

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    |           |E|R|R|R|N|N|N|N|C|C|R|R|S|R|T|E|C|S|S|S|S|S|S|S|S|C| |
    |           |E|M|M|M|P|P|P|P|L|L|T|T|T|S|E|S|L|R|R|R|R|R|R|R|R|D|
    |           |W|S|S|S|E|E|S|R|R|E|E|S|T|S|P|R|S|S|S|S|E|E|E|E|D|I|
    |           |E|L|L|L|1|0|T|T|E|B|S|S|E|D|T|E|O|T|T|T|T|3|2|1|0|I|S|
    |           |N|2|1|0| | |1|0| | |1|0|L| |S| | |3|2|1|0| | | |S| |
    +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    
```

Bit Descriptions:  
 =====

BIT<31:27>: NI0 ( Not implemented, Read as zero )

BIT<26>: EEPROM Write Enable, 1 = Enabled, 0 = Disabled  
 This bit enables the EEPROM Write Strobe to function. Its purpose is as a write protect against un-intended EEPROM Write Accesses. With this bit set to "0" or disabled, a write to EERPOM will appear to function correctly with the exception of the EEWE L control signal to the EEPROMs not asserting and de-asserting. The current cycle's data on the EEPROM Data Bus will not be written into the EEPROM. For firmware to perform writes to EEPROM, code must first enable this bit before writing to EEPROM. At reset, this bit defaults to a "0" or Function Disabled. This bit is READ/WRITE accessible.

BIT<25:23>: Reset Mux Select <2:0>, 1 = Enabled, 0 = Disabled  
 These bits are for use in Gate Array Fault Grading and should NOT be programmed during Self Test or Normal Operation. The default value will be read as all 0's. The following shows mux select codes vrs. load data for the Reset Counter. These values are only loaded into the Reset Counter following the cycling of INIT L provided RTEST0 ( GPCSR bit <15> ) is set to "1". At powerup, the bits are set to "0" selecting a reset counter timeout of 101msec.

| RMSL<2:0> | Reset Counter Load Value |
|-----------|--------------------------|
| =====     | =====                    |
| 000       | \$18 148E ( 101msec. )   |
| 001       | \$1F FFFF ( 134msec. )   |
| 010       | \$01 0000 ( 4.2msec. )   |
| 011       | \$00 0100 ( 16usec. )    |
| 100       | \$00 0001 ( 64nsec. )    |
| 101       | \$00 0007 ( 512nsec. )   |
| 110       | \$00 0230 ( 35usec. )    |
| 111       | \$00 0007 ( 512nsec. )   |

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

- BIT<22>: NP (CSR) Bus Even Parity, BYTE 1; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for NP (CSR) Bus Byte 1 Parity. The default value of this bit at logic "0" selects ODD Parity. This bit is Read/Write Access.  
This bit is for testing the Parity Protection Path between the PARSER and the AMI.
- BIT<21>: NP (CSR) Bus Even Parity, BYTE 0; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for NP (CSR) Bus Byte 0 Parity. The default value of this bit at logic "0" selects ODD Parity. This bit is Read/Write Access. This bit is for testing the Parity Protection Path between the PARSER and the AMI. This bit changes the parity checker to even. The parity generator is always odd.
- BIT<20>: NP (CSR) Bus Parity Error BYTE 1; 1 = Error, 0 = No Error  
This is a Parity Error Status Bit for Byte 1 or NPD[7:0] of the NP (CSR) Bus. This bit provides parity status to be used by Self Test Code to test Byte Parity. This status bit is read only and defaults to "0" on RESET. This bit changes the parity checker to even. The parity generator is always odd.
- BIT<19>: NP (CSR) Bus Parity Error BYTE 0; 1 = Error, 0 = No Error  
This is a Parity Error Status Bit for Byte 0 or NPD[15:8] of the NP (CSR) Bus. This bit provides parity status to be used by Self Test Code to test Byte Parity. This status bit is read only and defaults to "0" on RESET.
- BIT<18>: Clear Error Codes;  
Writing a 1 to this bit will generate a pulse that will clear the error codes sent to the ESP gate-array. This bit is write 1 only and read as 0. This bit defaults to "0" on RESET.
- BIT<17>: Clear Busy;  
Writing a 1 to this bit will generate a pulse that will clear the Busy Bits from the ESP gate-array. During Self Test, register accesses to the ESP Registers will not follow the ESP Bus Protocol and the ESP will not cycle the XMI Busy or PMC Busy bits correctly. This bit allows clearing out the AMI/ESP Interface Logic that is expecting the busy bits to be cycled correctly at this time. This bit is write 1 only and read as 0. This bit defaults to "0" on RESET.
- BIT<16>: RTEST1; 1 = Enabled, 0 = Disabled  
Reset Test 1 is a bit that when set enables the testing of the AMI Reset Logic during Fault Grading. This bit when set to a logic "1" will gang stages of the Reset Counter together for faster testing. This bit powers up in the logic "0" state or Disabled. This bit serves no purpose during Self Test or Normal Operation but is an assist to test during fault grading. This bit should never be set and will always read as a 0.
- BIT<15>: RTEST0; 1 = Enabled, 0 = Disabled  
Reset Test 0 is a bit that when set enables the testing of the AMI Reset Logic during Fault Grading. This bit when set to a logic "1" will block the INIT L signal from resetting the Reset Counter Mux Select Logic from its loaded value. This bit powers up in the logic "0" state or Disabled. This bit serves no purpose during Self Test or Normal Operation but is an assist to test during Fault Grading. This bit should never be set and will always read as a 0.
- BIT<14>: Self-Test Select; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", Enables the use of the Interrupt self-test register to generate interrupts. This bit is for testing the interrupt logic in the AMI gate-array. This bit is read/write and defaults to "0" on RESET.

DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY

- BIT<13>: Reset Disable; 1 = Enabled, 0 = Disabled  
This bit, when set to a logic "1", Enables the logic that WILL PREVENT a Node Reset from propagating through the AMI and resetting the 68020. This function is provided for the "warm resetting" of the Adapter. A time delay after the INIT L signal has been pulsed, logic will auto-clear this bit to logic "0" and Disable the Function.
- BIT<12>: TEST\_SEL H or Test Select; 1 = FDDI CSR Intfc, 0 = ESP Intfc Control  
This bit, is used to control a mux that gates buried control signals out of the AMI for testing purposes. The default value of this bit is logic "0" and selects ESP Control Signals. When set to a logic "1", it selects FDDI CSR BUS State Machine Bits. The control signals from the mux are brought out on TEST3,TEST2,TEST1,TEST0. This bit is Read/Write Access. This bit is used primarily for Fault Grading of the AMI.
- BIT<11>: ESP/AMI Interface Bus Even Parity Check; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking on AMI/ESP Bus reads. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access. This bit is for testing the Parity Protection Path on the AMI/ESP Bus.  
  
AMI/ESP Bus Parity Generation is always ODD Parity.
- BIT<10>: Clear Ownership; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1" will cause an automatic Clear Buffer Ownership to occur after a Buffer Descriptor Write in the AMI/ESP Interface's PM Direct Addressed mode of operation.  
  
When this bit is set to logic "0", no clear ownership will occur after a Buffer Descriptor Write. Logic "0" is the default setting of the bit at RESET. This bit is Read/Write.
- BIT<09>: SRAM Parity Error Byte 3; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 3 or D[7:0] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 3 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.  
| Write 1 to this bit to clear the parity error. This action clears  
| the PEAR register.
- BIT<08>: SRAM Parity Error Byte 2; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 2 or D[15:8] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 2 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.  
| Write 1 to this bit to clear the parity error. This action clears  
| the PEAR register.
- BIT<07>: SRAM Parity Error Byte 1; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 1 or D[23:15] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 1 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.  
| Write 1 to this bit to clear the parity error. This action clears  
| the PEAR register.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

- BIT<06>: SRAM Parity Error Byte 0; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 0 or D[31:23] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 0 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.  
| Write 1 to this bit to clear the parity error. This action clears  
| the PEAR register.
- BIT<05>: SRAM Even Parity Byte 3; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for Byte 3 or D[7:0] of SRAM. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access.
- BIT<04>: SRAM Even Parity Byte 2; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for Byte 2 or D[15:8] of SRAM. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access.
- BIT<03>: SRAM Even Parity Byte 1; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for Byte 1 or D[23:16] of SRAM. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access.
- BIT<02>: SRAM Even Parity Byte 0; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for Byte 0 or D[31:24] of SRAM. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access.
- BIT<01>: SRAM Parity Reset Disable; 1 = Parity Error Reset Enabled,  
0 = Parity Error Reset Disabled  
This bit when set to a logic "1" will enable SRAM Parity Errors to reset the 68020.  
If this bit is "1", writing to the SRAM Even Parity Byte<3:0> of this register ( bits <5:2> ) will not cause Even Parity checking. If this bit is "0", one can force even parity checking as a part of Self Test by writing GPCSR SRAM Even Parity Byte<3:0> ( bits <5:2> ). This bit must be set to a "0" to clear the SRAM Parity Error bits of GPCSR (bits <9:6>). This mode is intended for Self Test use only. The default value of this bit is logic "0"  
Disabling Parity Errors from propagate through to the reset logic. This bit is Read/Write Access.
- BIT<00>: Cache Disable; 1 = Cache Enabled, 0 = Cache Disabled  
This bit when set to a logic "1" will enable the 68020's internal cache. When set to a logic "0" will disable the 68020's internal cache. This bit is provided for debug purposes. The default setting of the bit is logic "0" at RESET or Cache Disabled. This bit is Read/Write Access.

## **5.4 AMI EVENT STROBES**

### **5.4.1 ADAPTER MANAGER TRANSMIT DONE STROBE (AMTDS)**

Adapter Manager ( 68020 ) Physical Address: \$0010 1010  
 =====

Access: SC ( Special Case - see bit descriptions )  
 =====

Initialized as: Not Applicable  
 =====

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
    +--+
    |0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
    +--+
    
```

Bit Descriptions:  
 =====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
 A longword write of any data to this address will generate the Adapter Manager Transmit Done ( AMXDN ) strobe. AMXDN is a signal to the PMC Gate Array that notifies it that the Adapter Manager has taken a packet off the Adapter Manager's Transmit Ring. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

**5.4.2 ADAPTER MANAGER RECEIVE DONE STROBE (AMRDS)**

Adapter Manager ( 68020 ) Physical Address: \$0010 1014  
 =====

Access: SC ( Special Case - see bit descriptions )  
 =====

Initialized as: Not Applicable  
 =====

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
    +--+
    |0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
    +--+
    
```

Bit Descriptions:  
 =====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
 A longword write of any data to this address will generate the Adapter Manager Receive Done ( AMRDN ) strobe. AMRDN is a signal to the PMC Gate Array that notifies it that the Adapter Manager has put a packet on the Adapter Manager's Receive Ring. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

**5.4.3 SMT PRIORITY STROBE (SMTPS)**



**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY**

Adapter Manager ( 68020 ) Physical Address: \$0010 1018  
=====

Access: SC ( Special Case - see bit descriptions )  
=====

Initialized as: Not Applicable  
=====

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
A longword write of any data to this address will generate the SMT\_PRIORITY interrupt. This strobe is used by SMT Code to enable the pre-emption of SMT Processing by generating an interrupt. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

**5.4.4 RTOS EVENT STROBE (RTOSES)**

Adapter Manager ( 68020 ) Physical Address: \$0010 101C  
=====

Access: SC ( Special Case - see bit descriptions )  
=====

Initialized as: Not Applicable  
=====

```

 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
A longword write of any data to this address will generate the RTOS\_EVENT interrupt. This interrupt is used by RTOS to assist in the scheduling of tasks. The act of writing to this address decodes the strobe; the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting. The Interrupt Priority Level (IPL) is 1.

**5.4.5 WATCHDOG TIMER CLEAR STROBE (WDTCLR)**

Adapter Manager ( 68020 ) Physical Address: \$0010 0C0C  
 =====

Access: SC ( Special Case - see bit descriptions )  
 =====

Initialized as: Not Applicable  
 =====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

Bit Descriptions:  
 =====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
 A longword write of any data to this address will generate the Watchdog Timer Clear. This strobe is used by the Watchdog Counter logic to clear the current count and reload the value of WDTIMR Register into the counter. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

## 5.5 AMI MISCELLANEOUS REGISTERS

### 5.5.1 PARITY ERROR ADDRESS REGISTER (PEAR)

Adapter Manager ( 68020 ) Physical Address: \$0010 1200  
 =====

Access: RO ( Read Only )  
 =====

INITIALIZED AS: \$00FF FFFF for power up reset.  
 =====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |0|0|0|0|0|0|0|0|          PE ADDRESS          |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

BIT<31:24>: NIO ( Not implemented, read as zero )  
 BIT<23:00>: Unless all set to logic "high", this longword will contain the SRAM memory address of the last SRAM Parity Error to occur.

Note: To clear this register SRAM Parity Reset Disable (BIT<01> of GPCSR) must be set to a 0, and one of the SRAM Parity Error Byte bits (BIT<09:06> of GPCSR) must be written to a 1. Clearing the error via bits 6-9 of the GPCSR register will clear the PEAR register.

## 5.6 INTERRUPT AND INTERRUPT VECTOR REGISTERS

### 5.6.1 INTERRUPT CSR 1 (INTCSR1)

Adapter Manager ( 68020 ) Physical Address: \$0010 1400

| Access: RW ( Read / Write 1 to clear)

INITIALIZED AS: All bits zero.

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|NIO|NIO|NIO|NIO|NIO|IPK|IPJ|IPI|IPH|IPG|IPF|IPE|IPD|IPC|IPB|IPA|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

- BIT[31:27]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[26]: IPK-RESERVED INTERRUPT PENDING  
 A 1 in this bit location indicates that a RESERVED interrupt is pending. This bit is used to poll the RESERVED interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[25]: IPJ-ESP CHIP INTERRUPT PENDING  
 A 1 in this bit location indicates that an ESP CHIP interrupt is pending. This bit is used to poll the ESP CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[24]: IPI-PM START TRANSMIT INTERRUPT PENDING  
 A 1 in this bit location indicates that an PM START TRANSMIT interrupt is pending. This bit is used to poll the PM START TRANSMIT interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[23]: IPH-FDDI NP CSR BUS PARITY ERROR INTERRUPT PENDING  
 A 1 in this bit location indicates that an FDDI NP CSR BUS PARITY ERROR interrupt is pending. This bit is used to poll the FDDI NP CSR BUS PARITY ERROR interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[22]: IPG-FDDI ELM CHIP INTERRUPT PENDING  
 A 1 in this bit location indicates that an FDDI ELM CHIP interrupt is pending. This bit is used to poll the FDDI ELM CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.

- BIT[21]: IPF-FDDI MAC CHIP INTERRUPT PENDING  
A 1 in this bit location indicates that an FDDI MAC CHIP interrupt is pending. This bit is used to poll the FDDI MAC CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[20]: IPE-FDDI RMC CHIP INTERRUPT PENDING  
A 1 in this bit location indicates that an FDDI RMC CHIP interrupt is pending. This bit is used to poll the FDDI RMC CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[19]: IPD-SMT PRIORITY INTERRUPT PENDING  
A 1 in this bit location indicates that an SMT PRIORITY interrupt is pending. This bit is used to poll the SMT PRIORITY interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[18]: IPC-PARSER INTERRUPT PENDING  
A 1 in this bit location indicates that an PARSER interrupt is pending. This bit is used to poll the PARSER interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[17]: IPB-HARDWARE TIMER INTERRUPT PENDING  
A 1 in this bit location indicates that an HARDWARE TIMER interrupt is pending. This bit is used to poll the HARDWARE TIMER interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[16]: IPA-EVENT LEVEL MONITOR INTERRUPT PENDING  
A 1 in this bit location indicates that an EVENT LEVEL MONITOR interrupt is pending. This bit is used to poll the EVENT LEVEL MONITOR interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.

## 5.6.2 INTERRUPT CSR 2 (INTCSR2)

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Adapter Manager ( 68020 ) Physical Address: \$0010 1402  
 =====

Access: RO ( Read Only )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                                                                                                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                                                                                                 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |
| 1                                                                                                 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| NIO NIO NIO NIO NIO NIO NIO NIO NIO NIO IP6 IP5 IP4 IP3 IP2 IP1                                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Bit Descriptions:  
 =====

- BIT[31:22]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[21]: IP6-IPL6 PENDING  
 A 1 in this bit location indicates that an IPL6 interrupt is pending. This bit is used to poll IPL6 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.
- BIT[20]: IP5-IPL5 PENDING  
 A 1 in this bit location indicates that an IPL5 interrupt is pending. This bit is used to poll IPL5 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.
- BIT[19]: IP4-IPL4 PENDING  
 A 1 in this bit location indicates that an IPL4 interrupt is pending. This bit is used to poll IPL4 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.
- BIT[18]: IP3-IPL3 PENDING  
 A 1 in this bit location indicates that an IPL3 interrupt is pending. This bit is used to poll IPL3 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.
- BIT[17]: IP2-IPL2 PENDING  
 A 1 in this bit location indicates that an IPL2 interrupt is pending. This bit is used to poll IPL2 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.
- BIT[16]: IP1-IPL1 PENDING  
 A 1 in this bit location indicates that an IPL1 interrupt is pending. This bit is used to poll IPL1 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

NOTE: IPL7 IS NON-MASKABLE

### 5.6.3 SOURCE INTERRUPT MASK REGISTER (SIMR)

Adapter Manager ( 68020 ) Physical Address: \$0010 1404

Access: R/W ( Read / Write )

INITIALIZED AS: All bits zero.

```

    3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|NIO|NIO|NIO|NIO|NIO|RES|ESP|PMX|FCP|ELM|MAC|RMC|SMT|PAR|HWT|EVL|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    
```

Bit Descriptions:

- BIT[31:27]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[26]: RES-RESERVED  
 Writing a 1 to this bit location will mask RESERVED interrupts. Masking RESERVED will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[25]: ESP-MASK ESP CHIP INTERRUPT  
 Writing a 1 to this bit location will mask ESP CHIP interrupts. Masking ESP CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[24]: PMX-MASK PM START TRANSMIT INTERRUPT  
 Writing a 1 to this bit location will mask PM START TRANSMIT interrupts. Masking PM START TRANSMIT will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[23]: FCP-MASK FDDI CSR PARITY ERROR INTERRUPT  
 Writing a 1 to this bit location will mask FDDI CSR PARITY ERROR interrupts. Masking FDDI CSR PARITY ERROR will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[22]: ELM-MASK FDDI ELM CHIP INTERRUPT  
 Writing a 1 to this bit location will mask FDDI ELM CHIP interrupts. Masking FDDI ELM CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[21]: MAC-MASK FDDI MAC CHIP INTERRUPT  
 Writing a 1 to this bit location will mask FDDI MAC CHIP interrupts. Masking FDDI MAC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

- BIT[20]: RMC-MASK FDDI RMC CHIP INTERRUPT  
Writing a 1 to this bit location will mask FDDI RMC CHIP interrupts. Masking FDDI RMC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[19]: SMT-MASK SMT PRIORITY INTERRUPT  
Writing a 1 to this bit location will mask SMT PRIORITY interrupts. Masking SMT PRIORITY will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[18]: PAR-MASK PARSER INTERRUPT  
Writing a 1 to this bit location will mask PARSER interrupts. Masking PARSER will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[17]: HWT-MASK HARDWARE TIMER INTERRUPT  
Writing a 1 to this bit location will mask HARDWARE TIMER interrupts. Masking HARDWARE TIMER will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[16]: EVL-MASK EVENT LEVEL MONITOR INTERRUPT  
Writing a 1 to this bit location will mask EVENT LEVEL MONITOR interrupts. Masking EVENT LEVEL MONITOR will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

#### **5.6.4 IPL INTERRUPT MASK REGISTER (IPLIMR)**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Adapter Manager ( 68020 ) Physical Address: \$0010 1406  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                                                                                             |   |     |   |     |   |     |   |     |   |     |   |     |   |     |   |     |  |     |  |     |  |     |  |
|---------------------------------------------------------------------------------------------|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|---|-----|--|-----|--|-----|--|-----|--|
| 3                                                                                           | 3 | 2   | 2 | 2   | 2 | 2   | 2 | 2   | 2 | 2   | 1 | 1   | 1 | 1   |   |     |  |     |  |     |  |     |  |
| 1                                                                                           | 0 | 9   | 8 | 7   | 6 | 5   | 4 | 3   | 2 | 1   | 0 | 9   | 8 | 7   | 6 |     |  |     |  |     |  |     |  |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |     |   |     |   |     |   |     |  |     |  |     |  |     |  |
| NIO                                                                                         |   | NIO |   | NIO |   | NIO |   | NIO |   | NIO |   | IM6 |   | IM5 |   | IM4 |  | IM3 |  | IM2 |  | IM1 |  |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |     |   |     |   |     |   |     |   |     |   |     |   |     |   |     |  |     |  |     |  |     |  |

Bit Descriptions:  
 =====

- BIT[31:22]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[21]: IM6-MASK IPL6  
 Writing a 1 to this bit location will mask IPL6 interrupts. Masking IPL6 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.
- BIT[20]: IM5-MASK IPL5  
 Writing a 1 to this bit location will mask IPL5 interrupts. Masking IPL5 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.
- BIT[19]: IM4-MASK IPL4  
 Writing a 1 to this bit location will mask IPL4 interrupts. Masking IPL4 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.
- BIT[18]: IM3-MASK IPL3  
 Writing a 1 to this bit location will mask IPL3 interrupts. Masking IPL3 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.
- BIT[17]: IM2-MASK IPL2  
 Writing a 1 to this bit location will mask IPL2 interrupts. Masking IPL2 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.
- BIT[16]: IM1-MASK IPL1  
 Writing a 1 to this bit location will mask IPL1 interrupts. Masking IPL1 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

NOTE: IPL7 IS NON-MASKABLE



### 5.6.5 SOURCE INTERRUPT ENABLE REGISTER (SIER)

Adapter Manager ( 68020 ) Physical Address: \$0010 1408  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                                                                                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                                                                                           | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |
| 1                                                                                           | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| NIO NIO NIO NIO NIO RES ESP PMX FCP ELM MAC RMC SMT PAR HWT EVL                             |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Bit Descriptions:  
 =====

- BIT[31:27]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[26]: RES-RESERVED  
 Writing a 1 to this bit location will enable RESERVED interrupts. Disabling RESERVED will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[25]: ESP-ENABLE ESP CHIP INTERRUPT  
 Writing a 1 to this bit location will enable ESP CHIP interrupts. Disabling ESP CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[24]: PMX-ENABLE PM START TRANSMIT INTERRUPT  
 Writing a 1 to this bit location will enable PM START TRANSMIT interrupts. Disabling PM START TRANSMIT will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[23]: FCP-ENABLE FDDI CSR BUS PARITY ERROR INTERRUPT  
 Writing a 1 to this bit location will enable FDDI CSR BUS PARITY ERROR interrupts. Disabling FDDI CSR BUS PARITY ERROR will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[22]: ELM-ENABLE FDDI ELM CHIP INTERRUPT  
 Writing a 1 to this bit location will enable FDDI ELM CHIP interrupts. Disabling FDDI ELM CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[21]: MAC-ENABLE FDDI MAC CHIP INTERRUPT  
 Writing a 1 to this bit location will enable FDDI MAC CHIP interrupts. Disabling FDDI MAC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

- BIT[20]:** RMC-ENABLE FDDI RMC CHIP INTERRUPT  
 Writing a 1 to this bit location will enable FDDI RMC CHIP interrupts. Disabling FDDI RMC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[19]:** SMT-ENABLE SMT PRIORITY INTERRUPT  
 Writing a 1 to this bit location will enable SMT PRIORITY interrupts. Disabling SMT PRIORITY will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[18]:** PAR-ENABLE PARSER INTERRUPT  
 Writing a 1 to this bit location will enable PARSER interrupts. Disabling PARSER will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[17]:** HWT-ENABLE HARDWARE TIMER INTERRUPT  
 Writing a 1 to this bit location will enable HARDWARE TIMER interrupts. Disabling HARDWARE TIMER will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[16]:** EVL-ENABLE EVENT LEVEL MONITOR INTERRUPT  
 Writing a 1 to this bit location will enable EVENT LEVEL MONITOR interrupts. Disabling EVENT LEVEL MONITOR will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

**5.6.6 IPL INTERRUPT ENABLE REGISTER (IPLIER)**

Adapter Manager ( 68020 ) Physical Address: \$0010 140A

=====

Access: R/W ( Read / Write )

=====

INITIALIZED AS: All bits zero.

=====

|                                                                   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3                                                                 | 3   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 1   | 1   | 1   | 1   |     |
| 1                                                                 | 0   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   | 9   | 8   | 7   | 6   |     |
| +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| NIO                                                               | NIO | NIO | NIO | NIO | NIO | NIO | NIO | NIO | NIO | NIO | IE6 | IE5 | IE4 | IE3 | IE2 | IE1 |
| +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+ |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

Bit Descriptions:

=====

- BIT[31:22]:** NIO-NOT IMPLEMENTED, READ AS 0
- BIT[21]:** IE6-ENABLE IPL6 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL6 interrupts. Disabling IPL6 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

BIT[20]: IE5-ENABLE IPL5 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL5 interrupts. Disabling IPL5 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[19]: IE4-ENABLE IPL4 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL4 interrupts. Disabling IPL4 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[18]: IE3-ENABLE IPL3 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL3 interrupts. Disabling IPL3 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[17]: IE2-ENABLE IPL2 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL2 interrupts. Disabling IPL2 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[16]: IE1-ENABLE IPL1 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL1 interrupts. Disabling IPL1 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

**5.6.7 INTERRUPT SELF TEST REGISTER (INTST)**

Adapter Manager ( 68020 ) Physical Address: \$0010 140C

=====

Access: R/W ( Read / Write )

=====

INITIALIZED AS: All bits zero.

=====

|                                                                                             |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|---------------------------------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3                                                                                           | 3   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 1   | 1   | 1   | 1   |
| 1                                                                                           | 0   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   | 9   | 8   | 7   | 6   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| NIO                                                                                         | NIO | WDT | ACF | AEP | RES | ESP | PMX | FCP | ELM | MAC | RMC | SMT | PAR | HWT | EVL |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

Bit Descriptions:

=====

BIT[31:30]: NIO-NOT IMPLEMENTED, READ AS 0

DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY

- BIT[29]: WDT-TEST WATCHDOG TIMER INTERRUPT  
Writing a 1 to this bit location will force a WATCHDOG TIMER interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[28]: ACF-TEST AC POWER FAIL INTERRUPT  
Writing a 1 to this bit location will force a AC POWER FAIL interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[27]: AEP-TEST AMI/ESP BUS PARITY ERROR INTERRUPT  
Writing a 1 to this bit location will force a AMI/ESP BUS PARITY ERROR interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[26]: RES-RESERVED  
Writing a 1 to this bit location will force a RESERVED interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[25]: ESP-TEST ESP CHIP INTERRUPT  
Writing a 1 to this bit location will force a ESP CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

- BIT[24]:           PMX-TEST PM START TRANSMIT INTERRUPT  
Writing a 1 to this bit location will force a PM START TRANSMIT interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[23]:           FCP-TEST FDDI CSR BUS PARITY ERROR INTERRUPT  
Writing a 1 to this bit location will force a FDDI CSR BUS PARITY ERROR interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[22]:           ELM-TEST FDDI ELM CHIP INTERRUPT  
Writing a 1 to this bit location will force a FDDI ELM CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[21]:           MAC-TEST FDDI MAC CHIP INTERRUPT  
Writing a 1 to this bit location will force a FDDI MAC CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[20]:           RMC-TEST FDDI RMC CHIP INTERRUPT  
Writing a 1 to this bit location will force a FDDI RMC CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.

- BIT[19]: SMT-TEST SMT PRIORITY INTERRUPT  
Writing a 1 to this bit location will force a SMT PRIORITY interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select of GPCSR must be set first.
- BIT[18]: PAR-TEST PARSER INTERRUPT  
Writing a 1 to this bit location will force a PARSER interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[17]: HWT-TEST HARDWARE TIMER INTERRUPT  
Writing a 1 to this bit location will force a HARDWARE TIMER interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[16]: EVL-TEST EVENT LEVEL MONITOR INTERRUPT  
Writing a 1 to this bit location will force a EVENT LEVEL MONITOR interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.

## 5.6.8 VECTOR REGISTERS (VECREGS)

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

**WATCHDOG TIMER VECTOR REGISTER (WTVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1500  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| WATCHDOG TIMER VECTOR      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

- BIT[31:24]: WATCHDOG TIMER VECTOR  
This area is to be programmed with the vector that will be used for WATCHDOG TIMER interrupt.
- BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**AC POWER FAIL VECTOR REGISTER (ACPFVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1502  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| AC POWER FAIL VECTOR      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

- BIT[31:24]: AC POWER FAIL VECTOR  
This area is to be programmed with the vector that will be used for AC POWER FAIL interrupt.
- BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**AMI/ESP BUS PARITY ERROR VECTOR REGISTER (ABPEVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1504  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| 1 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |

```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| AMI/ESP BUS PARITY ERROR VECTOR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

Bit Descriptions:  
 =====

BIT[31:24]: AMI/ESP BUS PARITY ERROR VECTOR  
 This area is to be programmed with the vector  
 that will be used for AMI/ESP BUS PARITY ERROR interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

ESP CHIP VECTOR REGISTER (ECVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 1506  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|   |                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
|---|-----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|
| 3 | 3               | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
| 1 | 0               | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
|   |                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |
|   | ESP CHIP VECTOR |   |   |   |   |   |   |   |   |   |   | 0 |   | 0 |   | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |
|   |                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |

Bit Descriptions:  
 =====

BIT[31:24]: ESP CHIP VECTOR  
 This area is to be programmed with the vector  
 that will be used for ESP CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

PARSER VECTOR REGISTER (PVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 1508  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|   |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |   |  |   |  |   |  |   |  |   |
|---|---------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|---|--|---|--|---|--|---|--|---|--|---|
| 3 | 3             | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |  |   |  |   |  |   |  |   |  |   |  |   |
| 1 | 0             | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |  |   |  |   |  |   |  |   |  |   |  |   |
|   |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |   |  |   |  |   |  |   |  |   |
|   | PARSER VECTOR |   |   |   |   |   |   |   |   |   |   | 0 |   | 0 |   | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |  | 0 |
|   |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |  |   |  |   |  |   |  |   |  |   |  |   |

Bit Descriptions:  
 =====

BIT[31:24]: PARSER VECTOR  
 This area is to be programmed with the vector  
 that will be used for PARSER interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

FDDI ELM CHIP VECTOR REGISTER (FECVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 150A  
 =====

Access: R/W ( Read / Write )  
 =====



**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

INITIALIZED AS: All bits zero.

=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI ELM CHIP VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

=====

BIT[31:24]: FDDI ELM CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI ELM CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

FDDI MAC CHIP VECTOR REGISTER (FMCVR)

=====

Adapter Manager ( 68020 ) Physical Address: \$0010 150C

=====

Access: R/W ( Read / Write )

=====

INITIALIZED AS: All bits zero.

=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI MAC CHIP VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

=====

BIT[31:24]: FDDI MAC CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI MAC CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

FDDI RMC CHIP VECTOR REGISTER (FRCVR)

=====

Adapter Manager ( 68020 ) Physical Address: \$0010 150E

=====

Access: R/W ( Read / Write )

=====

INITIALIZED AS: All bits zero.

=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI RMC CHIP VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

=====

BIT[31:24]: FDDI RMC CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI RMC CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**106 AMI REGISTERS**

SMT PRIORITY VECTOR REGISTER (SMTPVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 1510  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                   | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |
| 1                   | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |
| SMT PRIORITY VECTOR |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
 =====

BIT[31:24]: SMT PRIORITY VECTOR  
 This area is to be programmed with the vector  
 that will be used for SMT PRIORITY interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

PM START TRANSMIT VECTOR REGISTER (PMSTVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 1512  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                        | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |
| 1                        | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |
| PM START TRANSMIT VECTOR |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
 =====

BIT[31:24]: PM START TRANSMIT VECTOR  
 This area is to be programmed with the vector  
 that will be used for PM START TRANSMIT interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

HARDWARE TIMER VECTOR REGISTER (HTVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 1514  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

```

    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   HARDWARE TIMER VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT[31:24]: HARDWARE TIMER VECTOR  
This area is to be programmed with the vector  
that will be used for HARDWARE TIMER interrupt.

BIT[23:16]: NI0-NOT IMPLEMENTED, READ AS 0

EVENT LEVEL MONITOR VECTOR REGISTER (EVLVVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1516  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

```

    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| EVENT LEVEL MONITOR VECTOR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT[31:24]: EVENT LEVEL MONITOR VECTOR  
This area is to be programmed with the vector  
that will be used for EVENT LEVEL MONITOR interrupt.

BIT[23:16]: NI0-NOT IMPLEMENTED, READ AS 0

FDDI PARITY ERROR VECTOR REGISTER (FPEMVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1518  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

```

    3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1
    1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| FDDI PARITY ERROR VECTOR  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT[31:24]: FDDI PARITY ERROR VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI PARITY ERROR interrupt.

BIT[23:16]: NI0-NOT IMPLEMENTED, READ AS 0

**FDDI RESERVED VECTOR REGISTER (FRVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 151A  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                                                                                  |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|----------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                                                                                | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |
| 1                                                                                | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |
| +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+                |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| FDDI RESVERED VECTOR   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+                |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

**Bit Descriptions:**  
=====

- BIT[31:24]: FDDI RESERVED VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI RESERVED interrupt.
- BIT[23:16]: NI0-NOT IMPLEMENTED, READ AS 0

# CHAPTER 6

## PMC REGISTERS

This chapter describes all the registers available within the PMC gate array. They are visible only by the 68020.

Conventions:

All address values stated are the offset from the PMC CSR base address.

All undefined bits are NIO (Not Implemented, read as 0)

All register bits are write 1 to set, write 0 to clear unless otherwise stated.

Accessing a non-existent register in PMC CSR register block will return the value of the register accessed by the low 5 binary bits of the CSR accessed and cause an interrupt, unless masked.

### 6.1 PMC REGISTER SUMMARY

Following is a summary of the PMC CSR REGISTER BLOCK:

**Table 6: PMC CSR REGISTER ALLOCATION**

| ADDR             | BITS   | NAME                                      | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|--------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>CSR BLOCK</i> |        |                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ^X00             | <31:0> | PMC Control and Status Register           | This is the register of primary control for the PMC. See full definition elsewhere in this chapter.                                                                                                                                                                                                                                                                                                                                                                 |
| ^X01             | <31:0> | PMC RAM Parity Error Register             | In the event of a MIF parity error (see definition in CSR00), this register contains the address of the first detected address with data in error plus information identifying which parity protected data field(s) was in error. Under non-error circumstances this register is updated with the address of every read operation. This register can be written (for test purposes) only when REM_ENABLE is NOT set. See full definition elsewhere in this chapter. |
| ^X02             | <31:0> | PMC Interrupt and Interrupt Mask Register | In the event of a PMC interrupt, this register provides the reason(s) for the interrupt. All interrupts are maskable; the mask bits are in this register. See full definition elsewhere in this chapter.                                                                                                                                                                                                                                                            |
| ^X03             | <31:0> | PMC REM Protocol Error Register           | In the event that bit 4 of CSR02 is set (REM Protocol Error Summary Bit), this register provides full details of which protocol error(s) the REM encountered. See full definition elsewhere in this chapter.                                                                                                                                                                                                                                                        |

**Table 6 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR                        | BITS    | NAME                         | DEFINITION                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|---------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>REM RMC QUEUES BLOCK</i> |         |                              |                                                                                                                                                                                                                                                                                                                                                             |
| ^X04                        | <14:0>  | RMC_RCV_FIL<br>POINTER       | AM programmed as the base of the RMC_RCV ring, read as current RMC_RCV_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                          |
| ^X04                        | <30:16> | RMC_RCV_FREE<br>POINTER      | AM programmed as the first location in the RMC_RCV ring at which free space should be put by the REM (eg: RMC_RCV ring is 1024 entries long and is initialized with 500 buffers. RMC_RCV_FREE pointer is initialized at the 501st location). Read as current RMC_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                           |
| ^X04                        | <31:0>  | PMT Data Register 0          | This register doubles as the Packet Memory Test Data Register 0. It initializes as ^XAAAAAAAA and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                                                   |
| ^X05                        | <14:0>  | RMC_XMT_FIL<br>POINTER       | AM programmed as the base of the RMC_XMT ring, read as current RMC_XMT_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                          |
| ^X05                        | <30:16> | RMC_XMT_FREE<br>POINTER      | AM programmed as the base of the RMC_XMT ring, read as current RMC_XMT_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                         |
| ^X05                        | <31:0>  | PMT Data Register 4          | This register doubles as the Packet Memory Test Data Register 4. It initializes as ^X33333333 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                                                   |
| ^X06                        | <14:0>  | RMC_RCV_SIZE                 | AM programmed with the last address of the RMC_RCV ring and can only be written when REM_ENABLE is NOT set. The number of entries in the RMC_RCV ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                                              |
| ^X06                        | <30:16> | RMC_XMT_SIZE                 | AM programmed with the last address of the RMC_XMT RING and can only be written when REM_ENABLE is NOT set. The number of entries in the RMC_XMT ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                                              |
| ^X07                        | <31:0>  | PMT Failing Address Register | In the event that Packet Memory Test fails, this register will contain the failing address and some other information. While PMT is active, this register contains the current access address. See full definition elsewhere in this chapter. Initialized as 0 but may not be seen at this value since power-up PMT or programmed PMT may change this value |
| <i>REM AM QUEUES BLOCK</i>  |         |                              |                                                                                                                                                                                                                                                                                                                                                             |
| ^X08                        | <14:0>  | AM_RCV_FIL<br>POINTER        | AM programmed as the base of the AM_RCV ring, read as current AM_RCV_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                            |
| ^X08                        | <30:16> | AM_RCV_FREE<br>POINTER       | AM programmed as the first location in the AM_RCV ring at which free space should be put by the REM. Read as current AM_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                    |

**Table 6 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR | BITS    | NAME                      | DEFINITION                                                                                                                                                                                                                                                                                                                          |
|------|---------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X08 | <31:0>  | PMT Data Register 1       | This register doubles as the Packet Memory Test Data Register 1. It initializes as ^X55555555 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                           |
| ^X09 | <14:0>  | AM_XMT_FIL POINTER        | AM programmed as the base of the AM_XMT ring, read as current AM_XMT_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                    |
| ^X09 | <30:16> | AM_XMT_FREE POINTER       | AM programmed as the base of the AM_XMT ring, read as current AM_XMT_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                   |
| ^X09 | <31:0>  | PMT Data Register 5       | This register doubles as the Packet Memory Test Data Register 5. It initializes as ^X00000000 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                           |
| ^X0A | <14:0>  | AM_RCV_SIZE               | AM programmed with the last address of the AM_RCV ring and can only be written when REM_ENABLE is NOT set. The number of entries in the AM_RCV ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                        |
| ^X0A | <30:16> | AM_XMT_SIZE               | AM programmed with the last address of the AM_XMT ring and can only be written when REM_ENABLE is NOT set. The number of entries in the AM_XMT ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                        |
| ^X0B | <31:0>  | PMT Failing Data Register | In the event that Packet Memory Test fails, this register will contain the failing data pattern. While PMT is active, this register contains the current data pattern. See full definition elsewhere in this chapter. Initialized as 0 but may not be seen at this value since power-up PMT or programmed PMT may change this value |

*REM HOST QUEUES BLOCK*

|      |         |                       |                                                                                                                                                                                                                           |
|------|---------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X0C | <14:0>  | HOST_RCV_FIL POINTER  | AM programmed as the base of the HOST_RCV ring, read as current HOST_RCV_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                      |
| ^X0C | <30:16> | HOST_RCV_FREE POINTER | AM programmed as the first location in the HOST_RCV ring at which free space should be put by the REM. Read as current HOST_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                              |
| ^X0C | <31:0>  | PMT Data Register 2   | This register doubles as the Packet Memory Test Data Register 2. It initializes as ^X99999999 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail |
| ^X0D | <14:0>  | HOST_XMT_FIL POINTER  | AM programmed as the base of the HOST_XMT ring, read as current HOST_XMT_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                      |
| ^X0D | <30:16> | HOST_XMT_FREE POINTER | AM programmed as the base of the HOST_XMT ring, read as current HOST_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                     |

**Table 6 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR | BITS    | NAME                            | DEFINITION                                                                                                                                                                                                                                                                |
|------|---------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X0D | <31:0>  | PMT Data Register 6             | This register doubles as the Packet Memory Test Data Register 6. It initializes as ^X00000000 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                 |
| ^X0E | <14:0>  | HOST_RCV_SIZE                   | AM programmed with the last address of the HOST_RCV ring and can only be written when REM_ENABLE is NOT set. The number of entries in the HOST_RCV ring must be a power of 2. Can also be read. Initialized as 0                                                          |
| ^X0E | <30:16> | HOST_XMT_SIZE                   | AM programmed with the last address of the HOST_XMT ring and can only be written when REM_ENABLE is NOT set. The number of entries in the HOST_XMT ring must be a power of 2. Can also be read. Initialized as 0                                                          |
| ^X0F | <31:0>  | PMT Control and Status Register | This register controls and provides some status concerning the Packet Memory Test. See full definition elsewhere in this chapter. Initialized as ^X04000011 at power up only but may not be seen at this value since power-up PMT or programmed PMT may change this value |

*ALLOCATION REGISTERS*

|      |         |                              |                                                                                                                                                                                                                                                                                                                          |
|------|---------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X10 | <14:0>  | RMC_HOST ALLOCATION REGISTER | Written with the total number of RMC_FREE buffers to be allocated to the RMC/HOST path and can only be written when REM_ENABLE is NOT set. Read as the current count of available RMC to Host buffers. Initialized as 0                                                                                                  |
| ^X10 | <30:16> | RMC_AM ALLOCATION REGISTER   | Written with the total number of RMC_FREE buffers to be allocated to the RMC/AM path and can only be written when REM_ENABLE is NOT set. Read as the current count of available RMC to AM buffers. Initialized as 0                                                                                                      |
| ^X11 | <7:0>   | RMC_SMT OVERDRAFT REGISTER   | Written with the total number of RMC_FREE buffers to be allocated as an overdraft for the RMC/AM path, specifically for SMT packets and only when RMC_AM does not have enough buffers and can only be written when REM_ENABLE is NOT set. Read as the current count of available SMT overdraft buffers. Initialized as 0 |
| ^X11 | <15:8>  | RMC_MOP OVERDRAFT REGISTER   | Written with the total number of RMC_FREE buffers to be allocated as an overdraft for the RMC/AM path, specifically for MOP packets and only when RMC_AM does not have enough buffers and can only be written when REM_ENABLE is NOT set. Read as the current count of available MOP overdraft buffers. Initialized as 0 |
| ^X11 | <30:16> | RMC_ERR ALLOCATION REGISTER  | Written with the maximum number of error buffers that may be in circulation at any given time using RMC_FREE buffers and can only be written when REM_ENABLE is NOT set. Read as the current count of error buffers that may be delivered. Initialized as 0                                                              |

*FREE BUFFER COUNTERS*



**Table 6 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR                             | BITS    | NAME                   | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|---------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X12                             | <30:16> | HOST_FREE COUNTER      | AM programmed as number of buffers allocated at initialization to the HOST_RCV queue. Read as the current count of free buffers available for the HOST_RCV queue. Can only be written when REM_ENABLE is NOT set. Initialized as 0                                                                                                                                                                                                                    |
| <i>TRAFFIC CONTROL REGISTERS</i> |         |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| ^X13                             | <7:0>   | RMC backed up variable | If the RMC interface sees this number of RMC back-to-back requests, the ESP interface will be denied service. These bits are WRITE 1 TO SET AND WRITE 1 TO CLEAR and initialized as ^XFF                                                                                                                                                                                                                                                              |
| ^X13                             | <15:8>  | ESP lockout variable   | If the RMC interface considers the RMC to be backed up, the ESP will be denied service for this number of RMC back-to-back requests, before the ESP will be allowed service. These bits are WRITE 1 TO SET AND WRITE 1 TO CLEAR and initialized as ^XFF                                                                                                                                                                                               |
| <i>STATISTICS COUNTERS</i>       |         |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| ^X14                             | <23:0>  | PARSER DISCARD COUNTER | This counter contains a count of the packets discarded because the PARSER set the discard bit (ie packets not destined for this station). The count is cleared by being read. This counter will interrupt on setting of bit 23 (maskable). This will happen approximately every 10 secs in the case of an infinite stream of minimum sized packets all not for this station. This counter can be written when REM_ENABLE is not set. Initialized as 0 |
| ^X15                             | <15:0>  | HOST DISCARD COUNTER   | This counter contains a count of the HOST packets discarded because no RMC/HOST buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 15 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized LLC packets all destined for the host. This counter can be written when REM_ENABLE is not set.                                              |
| ^X15                             | <31:16> | AM DISCARD COUNTER     | This counter contains a count of the AM (but not SMT or MOP) packets discarded because no RMC/AM buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 31 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized LLC packets all destined for the adapter manager. This counter can be written when REM_ENABLE is not set.                  |
| ^X15                             | <31:0>  | PMT Data Register 3    | This register doubles as the Packet Memory Test Data Register 3. It initializes as ^XCCCCCCCC and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                                                                                                                                             |
| ^X16                             | <15:0>  | SMT DISCARD COUNTER    | This counter contains a count of the SMT packets discarded because no SMT overdraft buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 15 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized SMT packets all for this station. This counter can be written when REM_ENABLE is not set. Initialized as 0.                             |

**Table 6 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR | BITS    | NAME                            | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------|---------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X16 | <31:16> | MOP DISCARD COUNTER             | This counter contains a count of the MOP packets discarded because no MOP overdraft buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 31 (maskable). This will happen approximately every 100 mSecs in the case of an infinite stream of minimum sized MOP packets all for this station. This counter can be written when REM_ENABLE is not set. Initialized as 0.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ^X17 | <15:00> | OVERSIZE PACKET DISCARD COUNTER | This counter contains the count of HOST DESTINED packets discarded because the byte count in the RMC delivered Buffer Descriptor exceeded the value in the MAXIMUM SIZE REGISTER. This counter is cleared by reading this register. This counter will interrupt on setting bit 15 (maskable). This will happen approximately every 2.7 secs in the case of an infinite stream of minimum sized oversize packets all for this station. This counter can be written when REM_ENABLE is not set.                                                                                                                                                                                                                                                                                                                                                                       |
| ^X17 | <31:16> | ERROR DISCARD COUNTER           | This read-only counter contains a count of the ERROR packets discarded because either there were no AM buffers available or the ERROR ALLOCATION has been exceeded. The count is cleared by being read. This counter will interrupt on setting of bit 31 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized error packets all for this station. This counter can be written when REM_ENABLE is not set.                                                                                                                                                                                                                                                                                                                                                                                                   |
| ^X17 | <31:0>  | PMT Data Register 7             | This register doubles as the Packet Memory Test Data Register 7. It initializes as ^X0003C95A and can only be written when REM_ENABLE is NOT set. All PMT data registers other than this one contain the data for bits <31:0> of the DRAM; this register contains the data for bits <35:32> and each nibble is associated with one of the other data registers. The lowest nibble is associated with data register 0, the second lowest nibble with data register 1 and so on to the second highest nibble. The highest nibble is always read as 0 and not associated with any data. As an example to make the zeroth pattern all 1's one would write PMT data register 0 (CSR04) with ^X11111111 and PMT data register 7 (CSR17) bits <3:0> with ^B0001 leaving the other data unchanged. Note that if this register is written while PMT is active, PMT may fail. |
| ^X18 | <12:0>  | MAXIMUM SIZE REGISTER           | AM programmed as the maximum byte count that a host destined packet may have in order to be delivered. Packets exceeding this byte count will be counted and discarded. PLEASE NOTE THAT PACKETS OF ONE PAGE OR LESS CANNOT BE DISCARDED THIS WAY. This register can only be written when REM_ENABLE is NOT set. Initialized as 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| ^X19 | <31:0>  |                                 | Not implemented, read as ^X19191919.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ^X1A | <31:0>  |                                 | Not implemented, read as ^X1A1A1A1A.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ^X1B | <31:0>  |                                 | Not implemented, read as ^X1B1B1B1B                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1C | <31:0>  |                                 | Not implemented, read as ^X1C1C1C1C                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1D | <31:0>  |                                 | Not implemented, read as ^X1D1D1D1D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1E | <31:0>  |                                 | Not implemented, read as ^X1E1E1E1E                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1F | <31:0>  |                                 | Not implemented, read as ^X1F1F1F1F                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

## 6.2 PMC REGISTER DEFINITIONS

### 6.2.1 PMC CONTROL AND STATUS REGISTER (CSR00)

AM Read/Write (Read/Write - Write 1 to Toggle - Unless Otherwise Stated)

Initialized as ^X02000080

```

    BIT #      3          2 2          1 1          0 0          0
              1          4 3          6 5          8 7          0
    *-----v-----v-----v-----*
    |F|F|M|M|R|R|  |S|H|H|H|R|R|R|B|R|M|I|P|  |E|D|I|F|F|F|F|F|R|R| | |
    |P|P|M|E|E|E|REF|U|S|S|S|M|M|M|M|E|A|R|A|X|E|R|V|I|F|R|R|R|R|H|E|E|
    |B|B|E|M|F|F|  |R|T|T|T|C|C|C|R|M|S|X|S|  |N|S|V|M|M|R|R|S|M|M|
    |I|I|N|S|T|I|INT|M|R|T|S|R|T|S|K|F|K|W|P|COD|P|P|R|C|C|A|H|T|F|E|
    |X|R|A|I|S|N|  |O|D|D|T|D|D|T|R|T|R|B|A|  |A|A|D|X|R|M|S|A|R|N|
    |D|D|  |Z|T|I|1:0|D|N|N|X|N|N|X|Q|P|X|D|R|1:0|R|R|Y|D|D|  |T|M|Z|A|
    *-----^-----^-----^-----*
    
```

| Bit # | Mnemonic | Definition                                                                                                                                                                                                                                                                                                             |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | FPBIXD   | FORCE PBI TRANSMIT DONE: This bit is write only and read as 0. When written with a 1, it will issue a PBI transmit done to the REM as if it had come from PBI bus.                                                                                                                                                     |
| 30    | FPBIRD   | FORCE PBI RECEIVE DONE: This bit is bit is write only and read as 0. When written with a 1, it will issue a PBI receive done to the REM as if it had come from PBI bus.                                                                                                                                                |
| 29    | MMENA    | MEMORY MANAGEMENT ENABLE: This bit is readable and write 1 to set, write 1 to clear and initialized to 0. When asserted, all addresses associated with data or buffer descriptor transactions will be translated                                                                                                       |
| 28    | MEMSIZ   | MEMORY SIZE: This bit is read only. This bit reflects the assertion level of the MEMSIZ pin of the gate array. MEMSIZ=0 indicates the module has been manufactured with 1 MByte of DRAM and 16K Longwords of SRAM. MEMSIZ=1 indicates the module has been manufactured with 4 MByte of DRAM and 64K Longwords of SRAM. |
| 27    | REFTST   | REFRESH TEST: When asserted, enables the setting of REFINI [26]. It also replaces the AMSTX interrupt signal with an indication of refresh counter overflow.                                                                                                                                                           |
| 26    | REFINI   | REFRESH INIT: When this bit is asserted and refresh [27] is asserted, it will cause the DRAM refresh counters to initialize to all 0's.                                                                                                                                                                                |
| 25:24 | REFINT   | REFRESH INTERVAL: These bits are readable and write 1 to set, write 1 to clear and initialized to ^B10 (10.24uSec)                                                                                                                                                                                                     |
| 23    | SURMOD   | SURROGATE MODE: When asserted, RMC and ESP Transmit and Receive Dones are redirected via Adapter Manager Start Transmit Interrupt. In order to determine who caused the interrupt, read this register. Initialized as 0                                                                                                |
| 22    | HSTRDN   | HOST RECEIVE DONE: (Only meaningful when SURMOD is asserted.) When asserted, it indicates that one or more packets have been queued on the HOST RECEIVE RING. Writing 1 to this bit, clears. Writing 0 has no effect. Initialized as 0                                                                                 |
| 21    | HSTTDN   | HOST TRANSMIT DONE: (Only meaningful when SURMOD is asserted.) When asserted, it indicates that one or more packets have been transmitted from the HOST TRANSMIT RING. Writing 1 to this bit, clears. Writing 0 has no effect. Initialized as 0                                                                        |

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

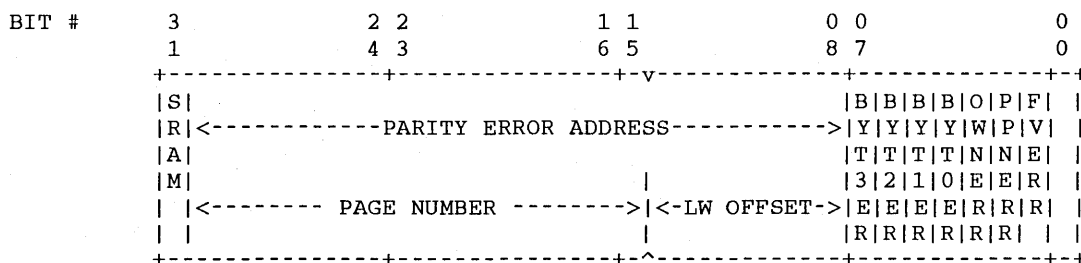
| Bit # | Mnemonic | Definition                                                                                                                                                                                                                                                                                                       |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20    | HSTSTX   | HOST START TRANSMIT: (Only meaningful when SURMOD is asserted.) When written with 1, a pulse is generated which indicates to the ESP that one or more packets have been queued on the HOST TRANSMIT RING. Writing 0 to this bit has no effect. Always read as 0, ie write only                                   |
| 19    | RMCRDN   | RMC RECEIVE DONE: (Only meaningful when SURMOD is asserted.) When asserted, it indicates that one or more packets have been queued on the RMC RECEIVE RING. Writing 1 to this bit, clears. Writing 0 has no effect. Initialized as 0                                                                             |
| 18    | RMCTDN   | RMC TRANSMIT DONE: (Only meaningful when SURMOD is asserted.) When asserted, it indicates that one or more packets have been transmitted from the RMC TRANSMIT RING. Writing 1 to this bit, clears. Writing 0 has no effect. Initialized as 0                                                                    |
| 17    | RM CSTX  | RMC START TRANSMIT: (Only meaningful when SURMOD is asserted.) When written with 1, a pulse is generated which indicates to the RMC that one or more packets have been queued on the HOST TRANSMIT RING. Writing 0 to this bit has no effect. Always read as 0, ie write only                                    |
| 16    | BMRKRQ   | BOOKMARK REQUEST: This is a Write Only bit, read as 0. When written with 1, a pulse is generated which starts the Bookmark State machine. The bookmark state machine ensures that the RMC_RCV queue is flushed up to the packet current at the time of bookmark request and the issues an interrupt (see CSR02). |
| 15    | REMFTP   | REM FORCE TRANSMIT PRIORITY: A read only copy of the internal Force Transmit Priority signal generated by the RMC interface of the PMC.                                                                                                                                                                          |
| 14    | MASKRX   | MASK RMC XMT: When asserted it will mask the the RMCSTX signal normally asserted by the REM.                                                                                                                                                                                                                     |
| 13    | IRXWBD   | When asserted it will cause the RMC Interface to internally discard transmit-write buffer descriptor transactions. The RMC is responded to as if the transaction actually happened.                                                                                                                              |
| 12    | PASPAR   | PASS PARITY: This bit is readable and write 1 to set, write 1 to clear and initialized to 0. When asserted, it will force the passing of internally generated parity, odd or even according to bit 9 of this register, with data from the PBI bus regardless of the parity passed on that bus.                   |
| 11:10 | XERCOD   | This is a read only copy of the AMI fatal error lines. Note that assertion of either XERCOD line causes FINITL to assert. This resets the FDDI chip set and Parser.                                                                                                                                              |
| 9     | EVNPAR   | EVEN PARITY: When asserted, all parity generators and checkers should generate/check even parity                                                                                                                                                                                                                 |
| 8     | DISPAR   | DISABLE PARITY: When asserted, all parity checkers will be disabled, ie parity errors will not be reported.                                                                                                                                                                                                      |
| 7     | IFVRDY   | FORWARDING VECTOR READY: When asserted, RMC_RCV_CLR_OWN transactions will be honored without waiting for the PARSER to assert FVRDY.                                                                                                                                                                             |
| 6     | FRMCXD   | FORCE RMC TRANSMIT DONE: This bit is write only and read as 0. When written with a 1, it will issue an RMC transmit done to the REM as if it had come from the RMC bus.                                                                                                                                          |
| 5     | FRMCRD   | FORCE RMC RECEIVE DONE: This bit is write only and read as 0. When written with a 1, it will issue an RMC receive done to the REM as if it had come from the RMC bus.                                                                                                                                            |
| 4     | FRRAM    | FORCE RMC_RCV buffers to Adapter Manager                                                                                                                                                                                                                                                                         |
| 3     | FRRHST   | FORCE RMC_RCV buffers to Host                                                                                                                                                                                                                                                                                    |
| 2     | FHSTAM   | FORCE HOST_RCV buffers to Adapter Manager                                                                                                                                                                                                                                                                        |

| Bit # | Mnemonic | Definition                                                                                                                                                        |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | REMFZ    | REM FREEZE: When asserted will cause the REM to complete a current packet move and/or free buffer move and stall. REM will resume upon deassertion of this signal |
| 0     | REMENA   | REM Enable, ie enable the state machines that run the REM. REM parameters CAN ONLY BE WRITTEN when REMENA is deasserted                                           |

### 6.2.2 PMC RAM PARITY ERROR REGISTER (CSR01)

AM READ/WRITE if CSR00[0] = 0, ie, for test purposes only. Otherwise:

ACCESS: ADAPTER MANAGER (SC, cleared on INIT or READ)

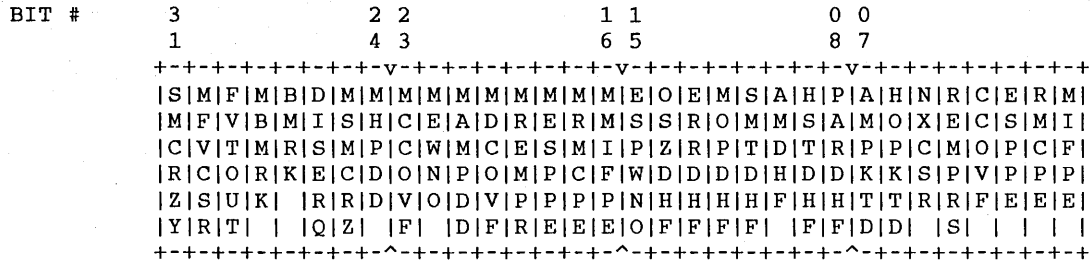


| Bit # | Mnemonic | Definition                                                                                                                                                         |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | SRAM     | If set, address is SRAM. If clear, address is DRAM.                                                                                                                |
| 30:8  | PERRAD   | Actual physical address of first detected error. Note that if a MIF parity error is not posted that this field contains the last accessed RAM read address         |
| 7     | BYT3ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 3, bits [31:24], has a parity error.     |
| 6     | BYT2ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 2, bits [23:16], has a parity error.     |
| 5     | BYT1ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 1, bits [15:8], has a parity error.      |
| 4     | BYT0ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 0, bits [7:0], has a parity error error. |
| 3     | OWNERR   | If the error is in a PTE, then this bit set indicates that the own field, bits [35:32], has a parity error.                                                        |
| 2     | PPNERR   | If the error is in a PTE, then this bit set indicates that the PPN field, bits [31:16], has a parity error.                                                        |
| 1     | FVERR    | If the error is in a PTE, then this bit set indicates that the FV field, bits [15:0], has a parity error.                                                          |
| 0     |          | Not implemented. Read as zero.                                                                                                                                     |

### 6.2.3 PMC MASK AND INTERRUPT REGISTER (CSR02)

ACCESS: ADAPTER MANAGER (SC, cleared on INIT )

Mask Bits are R/W toggle bits, ie, write 1 to set, write 1 to clear  
 Discard Counter 1/2 full bits are read only and cleared by reading  
 the relevant counter. They can only set if CSR00[0] = 1  
 Bits 27, 7:5 are read only and cleared by reading this register  
 Bits 29, 4:0 are read only and cleared by init only

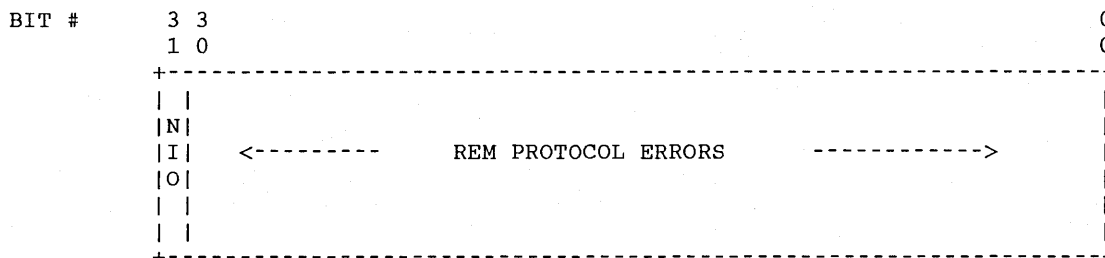


| Bit # | Mnemonic | Definition                                                                                                                                                                                                                                                  |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | SMCRZY   | STATE MACHINE UNKNOWN STATE : This condition causes an interrupt.                                                                                                                                                                                           |
| 30    | MFVCSR   | Mask forwarding vector timeout interrupt and access attempted to non existent CSR interrupt.                                                                                                                                                                |
| 29    | FVTOUT   | Forwarding Vector Timeout : This interrupt indicates that the PMC has waited 16 cycles since detecting an RMC_RCV clear own instruction and has not received FV Ready from the Parser Gate array.                                                           |
| 28    | MBMRK    | Mask Bookmark Interrupt.                                                                                                                                                                                                                                    |
| 27    | BMRK     | BOOKMARK : This interrupt indicates that the RMC_RCV queue has been flushed up to and including the packet current at the time of issuance of the Bookmark Request (See CSR00[16]).                                                                         |
| 26    | DISERQ   | DISABLE ESPRQENASM : This bit is readable and write 1 to set, write 1 to clear. When asserted, it will disable the affects of the ESP enable state machine. This will allow ESP requests to arbitrate for PMC memory regardless of possible RMC congestion. |
| 25    | MSMCRZ   | Mask state machine unknown state interrupt.                                                                                                                                                                                                                 |
| 24    | MHPDD    | Mask Host destined packet discarded interrupt.                                                                                                                                                                                                              |
| 23    | MCCOVF   | Mask CAS counter overflow interrupt.                                                                                                                                                                                                                        |
| 22    | MEWNO    | Mask ESP Write Not Owned interrupt.                                                                                                                                                                                                                         |
| 21    | MAMPDD   | Mask AM destined packet discarded interrupt.                                                                                                                                                                                                                |
| 20    | MDCOVF   | Mask Discard Counter Overflow Bits                                                                                                                                                                                                                          |
| 19    | MREMPR   | Mask Rem Protocol Error.                                                                                                                                                                                                                                    |
| 18    | MESPPE   | Mask ESP Parity Error.                                                                                                                                                                                                                                      |
| 17    | MRMCPE   | Mask RMC Parity Error.                                                                                                                                                                                                                                      |
| 16    | MMIFPE   | Mask MIF Protected Parity Error.                                                                                                                                                                                                                            |
| 15    | ESPWNO   | ESP Write Attempted, Not Owned. This condition causes the interrupt.                                                                                                                                                                                        |
| 14    | OSZDHF   | Oversize Discard Counter Half Full. This condition causes the interrupt.                                                                                                                                                                                    |
| 13    | ERRDHF   | Error Discard Counter Half Full. This condition causes the interrupt.                                                                                                                                                                                       |
| 12    | MOPDHF   | MOP Discard Counter Half Full. This condition causes the interrupt.                                                                                                                                                                                         |

| Bit # | Mnemonic | Definition                                                                        |
|-------|----------|-----------------------------------------------------------------------------------|
| 11    | SMTDHF   | SMT/MAC Discard Counter Half Full. This condition causes the interrupt.           |
| 10    | AMDHF    | AM Discard Counter Half Full. This condition causes the interrupt.                |
| 9     | HSTDHF   | HOST Discard Counter Half Full. This condition causes the interrupt.              |
| 8     | PARDHF   | PARSER Discard Counter Half Full. This condition causes the interrupt.            |
| 7     | AMPKTD   | AM Destined Packet Discarded. This condition causes the interrupt.                |
| 6     | HOPKTD   | HOST Destined Packet Discarded. This condition causes the interrupt.              |
| 5     | NXCSR    | Access Attempted to Nonexistent CSR Address. This condition causes the interrupt. |
| 4     | REMPRS   | REM Protocol Error Summary. This condition causes the interrupt.                  |
| 3     | CCOVF    | CAS Counter Overflowed. This condition causes the interrupt.                      |
| 2     | ESPPE    | Parity Error Detected at ESP Interface. This condition causes the interrupt.      |
| 1     | RMCPE    | Parity Error Detected at RMC Interface. This condition causes the interrupt.      |
| 0     | MIFPE    | Parity Error Detected in MIF. This condition causes the interrupt.                |

### 6.2.4 PMC PROTOCOL ERROR REGISTER (CSR03)

ACCESS: ADAPTER MANAGER (RO, cleared on INIT)



| Bit # | Mnemonic | Definition                                                                                                                        |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------|
| 31    | NIO      | Not implemented, read as 0                                                                                                        |
| 30    | PROE30   | Protocol Error 30: Three page or more Packet. Reading intermediate Page. EOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK.        |
| 29    | PROE29   | Protocol Error 29: Three page or more Packet. Reading intermediate Page. SOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK.        |
| 28    | PROE28   | Protocol Error 28: Two page or multiple page Packet. Reading last Page. EOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK          |
| 27    | PROE27   | Protocol Error 27: Two page or multiple page Packet. Reading last Page. SOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK.         |
| 26    | PROE26   | Protocol Error 26: Color Code indicates 00 (invalid) or 11 (to host). HOST to RMC or HOST to AM COPY. HOST MOVE FREE LOGIC BLOCK. |

This Register is READ ONLY and cleared only on INIT.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

| Bit # | Mnemonic | Definition                                                                                                                                                                               |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25    | PROE25   | Protocol Error 25: Color Code indicates 00 (invalid) or 01 (to RMC). RMC to AM or RMC to HOST COPY. RMC MOVE FREE LOGIC BLOCK. Please note that PROE24 and PROE25 are reversed from PMC1 |
| 24    | PROE24   | Protocol Error 24: Color Code indicates 00 (invalid) or 10 (to AM). AM to RMC or AM to HOST COPY. AM MOVE FREE LOGIC BLOCK. Please note that PROE24 and PROE25 are reversed from PMC1    |
| 23    | PROE23   | Protocol Error 23: Reading the first Page Table Entry from the HOST_RCV ring of a HOST generated Packet. HOST MOVE FILL LOGIC BLOCK.                                                     |
| 22    | PROE22   | Protocol Error 22: Reading the first Page Table Entry from the AM_RCV ring of an AM generated Packet. AM MOVE FILL LOGIC BLOCK.                                                          |
| 21    | PROE21   | Protocol Error 21: Reading the first Page Table Entry from the RMC_RCV ring of a RMC generated Packet. RMC MOVE FILL LOGIC BLOCK.                                                        |
| 20    | PROE20   | Protocol Error 20: Three page or more Packet. Reading intermediate Page. EOP=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                                  |
| 19    | PROE19   | Protocol Error 19: Three page or more Packet. Reading intermediate Page. SOP=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                                  |
| 18    | PROE18   | Protocol Error 18: Two page or multiple page Packet. Reading last Page. EOP=0. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                                   |
| 17    | PROE17   | Protocol Error 17: Two page or multiple page Packet. Reading last Page. SOP=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                                   |
| 16    | PROE16   | Protocol Error 16: Two page or greater Packet. OWN=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                                                            |
| 15    | PROE15   | Protocol Error 15: Two page or multiple page Packet. Reading last page. EOP=0. Multiple page Packet, reading intermediate page and EOP=1. AMRMCCOPY S.M. AM MOVE FILL LOGIC BLOCK.       |
| 14    | PROE14   | Protocol Error 14: Two page or multiple page Packet. Reading last page. SOP=1. Multiple page Packet, reading intermediate page and SOP=1. AMRMCCOPY S.M. AM MOVE FILL LOGIC BLOCK.       |
| 13    | PROE13   | Protocol Error 13: Two page or greater Packet. OWN=1. AMHOSTCOPY S.M. AM MOVE FILL LOGIC BLOCK.                                                                                          |
| 12    | PROE12   | Protocol Error 12: Two page or greater Packet. OWN=1. AMRMCCOPY S.M. AM MOVE FILL LOGIC BLOCK.                                                                                           |
| 11    | PROE11   | Protocol Error 11: Two page or greater Packet. OWN=1. HOSTAMCOPY S.M. HOST MOVE FILL LOGIC BLOCK.                                                                                        |
| 10    | PROE10   | Protocol Error 10: Two page or greater Packet. OWN=1. HOSTRMCCOPY S.M. HOST MOVE FILL LOGIC BLOCK.                                                                                       |
| 9     | PROE9    | Protocol Error 9: Two page or greater Packet. OWN=1. RMCHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                                                         |
| 8     | PROE8    | Protocol Error 8: Three page or more Packet. Reading intermediate Page. EOP=1. RMCHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                               |
| 7     | PROE7    | Protocol Error 7: Three page or more Packet. Reading intermediate Page. SOP=1. RMCHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                               |
| 6     | PROE6    | Protocol Error 6: Two page or multiple page Packet. Reading last Page. EOP=0. RMCHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                                |
| 5     | PROE5    | Protocol Error 5: Two page or multiple page Packet. Reading last Page. SOP=1. RMCHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                                |
| 4     | PROE4    | Protocol Error 4: Two page or greater Packet. OWN=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                                                           |

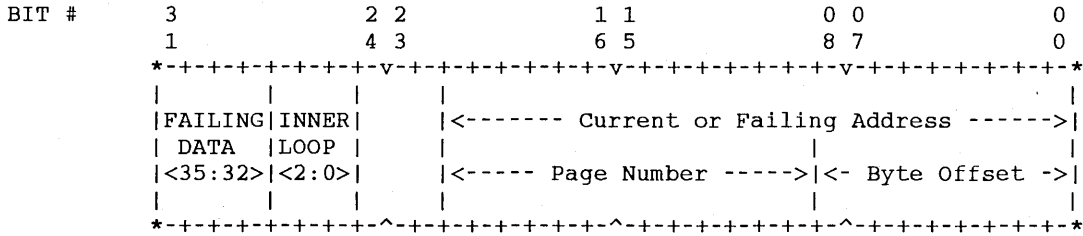


| Bit # | Mnemonic | Definition                                                                                                               |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------|
| 3     | PROE3    | Protocol Error 3: Three page or more Packet. Reading intermediate Page. EOP=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK. |
| 2     | PROE2    | Protocol Error 2: Three page or more Packet. Reading intermediate Page. SOP=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK. |
| 1     | PROE1    | Protocol Error 1: Two page or multiple page Packet. Reading last Page. EOP=0. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.  |
| 0     | PROE0    | Protocol Error 0: Two page or multiple page Packet. Reading last Page. SOP=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.  |

### 6.2.5 PMC PMT FAILING ADDRESS REGISTER (CSR07)

ACCESS: AM Read/Write (IFF CSR0F[0] = 0 else read only) - Cleared on INIT

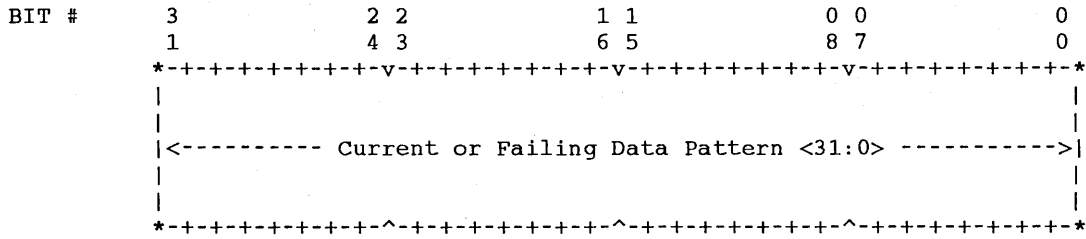
Bits 24:22 have no function. Read as 0 if CSR0F[0] = 1



| Bit #   | Mnemonic                   | Definition                                                                                                                                                                                       |
|---------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <31:28> | Failing Data <35:32>       | If PMT is active, this field contains the current data pattern or, if PMT has failed, the failing pattern or if passed, the last used data pattern                                               |
| <27:25> | Inner Loop <2:0>           | This counter indicates which pattern set is current. For example if this counter = 2 it would indicate the current pattern being used for test started with the contents of PMT Data Register 2. |
| <24:22> |                            | This bits have no function. They are read/writeable for test purposes only                                                                                                                       |
| <21:0>  | Current or Failing Address | If PMT is active, this field contains the current address or, if PMT has completed, the last accessed address.                                                                                   |

**6.2.6 PMC PMT FAILING DATA REGISTER (CSR0B)**

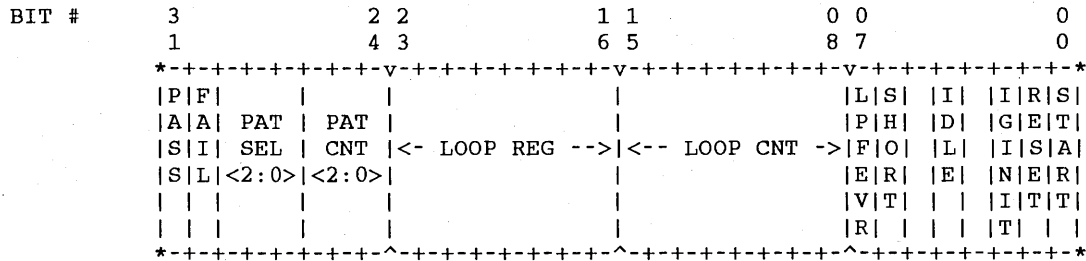
ACCESS: AM Read/Write (IFF CSR0F[0] = 0 else read only) - Cleared on INIT



| Bit #  | Mnemonic                       | Definition                                                                                                                                         |
|--------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <31:0> | Current or Failing Data <31:0> | If PMT is active, this field contains the current data pattern or, if PMT has failed, the failing pattern or if passed, the last used data pattern |

**6.2.7 PMC PMT CONTROL AND STATUS REGISTER (CSR0F)**

ACCESS: SC - INITIALIZED to ^X04000011 except bit 2 (see definition)



| Bit #   | Mnemonic      | Definition                                                                                                                                                                          |
|---------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <31>    | PASS          | When set this bit indicates that PMT was successful in testing the PMC DRAM. This bit is read only                                                                                  |
| <30>    | FAIL          | When set this bit indicates that PMT was NOT successful in testing the PMC DRAM. For further information refer to the PMT Failing address and Data registers. This bit is read only |
| <29:27> | PAT SEL <2:0> | This data field indicates which PMT data register is currently selected and being used. This field is read only                                                                     |

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

| Bit #   | Mnemonic       | Definition                                                                                                                                                                                                                                                                                                                                                                                   |
|---------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <26:24> | PAT CNT <2:0>  | This data field indicates how many patterns will be used for each loop of PMT where 0 = one pattern, 1 = two patterns and so on. This field is read/write where the bits are toggle bits, ie write 1 to set, write 1 to clear if CSROF[0] = 0 else read only.                                                                                                                                |
| <23:16> | LOOP REG <7:0> | This data field indicates which loop of PMT is currently being executed. This field is read only                                                                                                                                                                                                                                                                                             |
| <15:8>  | LOOP CNT <7:0> | This data field indicates how many loops will be used for PMT where 0 = one loop, 1 = two loops and so on. This field is read/write where the bits are toggle bits, ie write 1 to set, write 1 to clear if CSROF[0] = 0 else read only                                                                                                                                                       |
| <7>     | LPFEVR         | When set this bit causes PMT to run continuously. Such a test can be cleanly ended by resetting this bit and waiting for the normal end, ie LOOP CNT = LOOP REG. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable                                                                                                                         |
| <6>     | SHORT          | When set this bit causes PMT to apply patterns to only the first and last 8 longwords of memory for each pattern requested for test. This feature offers a fast way to stop PMT once started without initialization. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable                                                                     |
| <5>     |                | NIO                                                                                                                                                                                                                                                                                                                                                                                          |
| <4>     | IDLE           | This bit when asserted indicated that PMT is not active. This bit is read only                                                                                                                                                                                                                                                                                                               |
| <3>     |                | NIO                                                                                                                                                                                                                                                                                                                                                                                          |
| <2>     | IGINIT         | This bit when asserted will prevent PMT from starting in the event of a programmed init (ie a PMC init of 1 cycle). This bit is cleared on a power-up thus not preventing PMT from starting under such conditions. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable                                                                       |
| <1>     | RESET          | In the event of PMT failing, writing a 1 to this bit will allow the PMT state machine to go to its idle state. Note that PMT failing address and data registers should be read prior to setting this bit. This bit is write only and read as 0                                                                                                                                               |
| <0>     | START          | This bit when transitioned from deasserted to asserted will start PMT (if IGINIT = 0). This bit is set on initialization and is cleared by the PMT state machine upon completion of PMT. Note that deasserting this bit once PMT has started, ie, IDLE is deasserted, will not stop PMT. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable |

# CHAPTER 7

## PARSER REGISTERS AND DATABASE

This chapter describes all the registers available within the Parser gate array. They are visible only by the Adapter Manager (68020 Subsystem). All address values stated are the offset from the Parser CSR base address.

The CSRs contains control and status information about the Parser and priori knowledge about the whole system e.g the existence of Unknown user, AMC user etc. The Parser Database contains the complete filtering information including the list of enabled DAs, LLCs etc.

### 7.1 CSR REGISTERS

#### 7.1.1 CONTROL REGISTER 0 (CR0)

This control register contains the chip operating parameters and hardware testing hooks.

AM Access : R/W  
Internal (Parser) Access : R/W  
NP Address : 00 H  
Power Up Value : 0000 H

| 15 | 14  | 13 | 12  | 11 | 10   | 9   | 8   | 7   | 6   | 5 | 2    | 1 | 0   |
|----|-----|----|-----|----|------|-----|-----|-----|-----|---|------|---|-----|
| FR | DRP | RP | FVP | PS | FVRD | CBP | DBP | ACT | RMC |   | RADD |   | NIO |
|    |     |    |     |    |      |     |     |     | TM  |   |      |   |     |

BIT<15> : Force Reset = When this bit is "1", the Parser gate array gets reset. The reset includes the resetting of all CSR bits except this particular bit, all state machines and internal storage elements. The reset condition prevails for all other interfaces except the CSR interface, until a "0" is written to this bit. The Parser gate array needs complete reinitialization of its CSRs in order to restart. This bit is set to "0" after power up reset.

BIT<14> : Disable RMC Parity = When this bit is "1", the Parser disables the RMC bus parity checking. This bit is set to "0" after Parser reset, which is the normal mode of operation.

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

- BIT<13> : Odd/Even RMC Parity = When this bit is "1", the Parser assumes that the RMC is programmed to generate even parity ( parity is "0" if the number of HIGH data signals is even.) on its memory bus for write cycles. For more information, refer to RMC specs. This bit is set to "0" (odd parity) after Parser reset, which is the normal mode of operation.
- BIT<12> : Odd/Even Forwarding Vector Parity = When this bit is "1", the Parser generates even parity ( parity is "0" if the number of HIGH data signals is even.) for its forwarding vector. This bit is set to "0" (odd parity) after Parser reset, which is the normal mode of operation.
- BIT<11> : Parser Start = The adapter manager sets this bit to "1" after it has completely initialized the chip. When this bit is "1", the Parser starts functioning. This bit is reset to "0" after Parser reset.
- BIT<10> : Forwarding Vector Ready Disable = When this bit is set to "1" the forwarding vector ready signal to PMC gets disabled. This bit is reset to "0" after Parser reset.
- BIT<9> : Odd/Even CSR BUS Parity = When this bit is "1", the Parser checks for even parity ( parity is "0" if the number of HIGH data signals is even.) for its CSR Bus. This bit is set to "0" (odd parity) after Parser reset, which is the normal mode of operation.
- BIT<8> : Odd/Even Database BUS Parity = When this bit is "1", the Parser checks for even parity ( parity is "0" if the number of HIGH data signals is even.) for its Database Bus. This bit is set to "0" (odd parity) after Parser reset, which is the normal mode of operation.
- BIT<7> : Address Counter Test = This bit is used to test the Address Counter of the RAM Interface of the Parser. When this bit is set to "1", it breaks the 13-bit counter chain to count faster in stead of counting sequentially from 0 to 8K. This is bit is set to "0" after Parser reset.
- BIT<6> : RMC Test Mode = This bit is used to bring the Parser RMC Interface in the Test Mode. When this bit is set to "1", the test mode is entered. This bit is set to "0" after Parser reset. For more information, please see the Parser Design Specs.

BIT<5:1> : RMC Register Address =  
 These bits are used in conjunction with the bit <6>. When the Parser is in the RMC Interface test mode, these bits select the RMC Interface Registers to be examined. These bits are set to "0" after Parser reset. For more information, please see the Parser Design

Specs.

BIT<0> : Not Implemented, read as "0".

### 7.1.2 CONTROL REGISTER 1 (CR1)

This control register contains the initial operating parameters required by the Parsing Algorithm.

AM Access : R/W  
 Internal (Parser) Access : R only  
 NP Address : 01 H  
 Power Up Value : 0000 H

| 15  | 14  | 13  | 12   | 11   | 10   | 9    | 8    | 7    | 6   | 0 |
|-----|-----|-----|------|------|------|------|------|------|-----|---|
| UNK | AMC | NDE | Null | Null | IFCS | SNAP | SNAP | IFCS | NIO |   |
|     |     |     | NXTD | RXTD | NULL | NXTD | RXTD | SNAP |     |   |

BIT<15> : Unknown User = When this bit is "1", the Parser assumes an unknown user exists. This bit is set to "0" after the Parser reset. The unknown user index is not required by Parser as it always assumes it to be 31 (1F H).

BIT<14> : All Multicast User = When this bit is "1", the Parser assumes that at least one "All Multicast User" has been enabled. This bit is set to "0" after the Parser reset.

BIT<13> : Null DSAP Enabled = When this bit is "1", the Parser assumes that there is an user, who has enabled "00 H" as its DSAP. This bit is set to "0" after the Parser reset.

BIT<12> : Null dsap Non XID/Test packets Discard = When this bit is "1", the Parser discards the packet containing DSAP as NULL and CNTL as neither XID nor Test.

BIT<11> : Null dsap Response XID/Test packets Discard = When this bit is "1", the Parser discards the packet containing DSAP as NULL, SSAP as Response and CNTL as either XID or Test.

BIT<10> : Null DSAP Ignore FCS CRC Error = When this bit is "1", the Parser ignores the CRC error for the packet containing DSAP as NULL.

NOTE : These three bits <12:10> bring programmability for the packets having Null as DSAP and no user has enabled Null as its DSAP. For more details, please refer to the Parsing Algorithm.

BIT<09> : Snap dsap Non XID/Test packets Discard = When this bit is "1", the Parser discards the packet containing DSAP as SNAP and CNTL as neither XID nor Test.



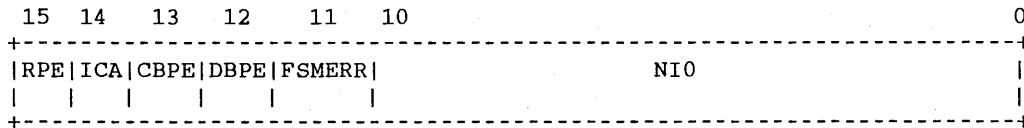
BIT<11> : Finite State Machine Error = When a state machine goes to an unknown state, this interrupt is generated. This bit tells the Adapter Manager that the cause of interrupt is the Problem in any one the state machines within the gate array.

BIT<10:0> : Not Implemented, read as "0".

### 7.1.3.2 INTERRUPT MASK REGISTER (IMR)

This register is used by the Adapter Manager to mask out the cause of the interrupt. A "0" in a particular bit position enables that cause of the interrupt and a "1" masks that out ie the bit in ICR is not updated and hence the interrupt is not generated. This register is reset by the Parser reset.

AM Access : R/W  
 Internal (Parser) Access : R only  
 NP Address : 03 H  
 Power Up Value : 0000 H



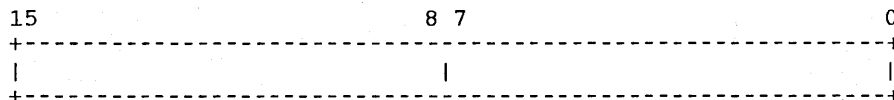
BIT<15> : RMC Parity Error  
 BIT<14> : Illegal CSR Access  
 BIT<13> : CSR Bus Parity Error  
 BIT<12> : Database Bus Parity Error  
 BIT<11> : Finite State Machine Error  
 BIT<10:0> : Not Implemented, read as "0".

### 7.1.4 DATABASE UPDATE REGISTER(S)

The following eight registers are used to update the Parser Database. The first four registers contain the 64-bit data, the fifth register contains the address of the RAM and the remaining three registers contain the control information.

(i) Data Register 3 (DR3)

AM Access : R/W  
 Internal (Parser) Access : R/W  
 NP Address : 04 H  
 Power Up Value : FFFF H



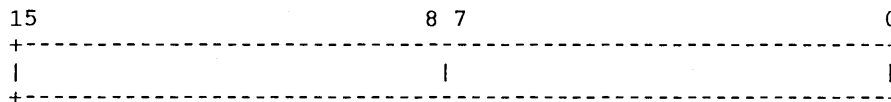
BIT<15:8> : Byte 7  
 BIT<7:0> : Byte 6  
 BIT<15:0> : Word 3  
 BIT<15:0> : MSW of Long Word 1  
 BIT<15:0> : MSW of Quad Word



**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

(ii) Data Register 2 (DR2)

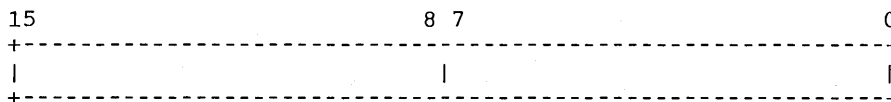
AM Access : R/W  
 Internal (Parser) Access : R/W  
 NP Address : 05 H  
 Power Up Value : FFFF H



BIT<15:8> : Byte 5  
 BIT<7:0> : Byte 4  
 BIT<15:0> : Word 2  
 BIT<15:0> : LSW of Long Word 1  
 BIT<15:0> : LSW of MSLW of Quad Word

(iii) Data Register 1 (DR1)

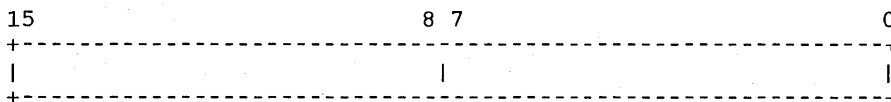
AM Access : R/W  
 Internal (Parser) Access : R/W  
 NP Address : 06 H  
 Power Up Value : FFFF H



BIT<15:8> : Byte 3  
 BIT<7:0> : Byte 2  
 BIT<15:0> : Word 1  
 BIT<15:0> : MSW of Long Word 0  
 BIT<15:0> : MSW of LSLW of Quad Word

(iv) Data Register 0 (DR0)

AM Access : R/W  
 Internal (Parser) Access : R/W  
 NP Address : 07 H  
 Power Up Value : FFFF H



BIT<15:8> : Byte 1  
 BIT<7:0> : Byte 0  
 BIT<15:0> : Word 0  
 BIT<15:0> : LSW of Long Word 0  
 BIT<15:0> : LSW of Quad Word

(v) Address Register (AR)



**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

BIT<11> : Read/Write = When this bit is "1", the update involves the read of the Database and when this bit is "0", the update involves the write to the Database. This bit is cleared to "0" on the Parser reset.

BIT<10:0> : Not Implemented, read as "0".

(vii) User Validity Register 1 (UVR1)

AM Access : R/W  
Internal (Parser) Access : R only  
NP Address : 0A H  
Power Up Value : 0000 H

```
15   0
+-----+
|User31|User30|   |User16|
+-----+
```

BIT<15> : User 31 = The "1" in this position show that the user number 31st is a valid user and all the packets received on his behalf should be forwarded. A "0" in this position invalidates that user and all packets for him are discarded. This bit is cleared to "0" on the Parser reset.

Similar description for all other bits.

(viii) User Validity Register 0 (UVR0)

AM Access : R/W  
Internal (Parser) Access : R only  
NP Address : 0B H  
Power Up Value : 0000 H

```
15   0
+-----+
|User15|User14|   |User0 |
+-----+
```

BIT<15> : User 15 = The "1" in this position show that the user number 15th is a valid user and all the packets received on his behalf should be forwarded. A "0" in this position invalidates that user and all packets for him are discarded. This bit is cleared to "0" on the Parser reset.

Similar description for all other bits.

## 7.2 PARSER DATABASE

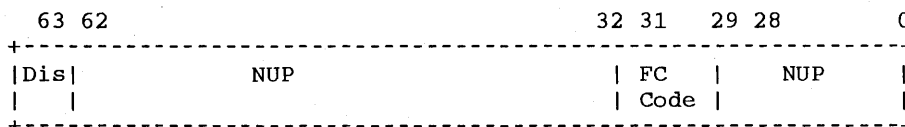
The Parser Database contains all the filtering information about the packets. Whether the packet should be discarded or forwarded, if forwarded, what is the destination, who is the eventual user etc., all these information are obtained from the Parser Database. This makes the Parser Subsystem implementation very flexible.

Updating the CAM does not interfere with the process of address matching for the MAC. Update of the CAM and of the Parser databases with DA information should be done separately. It is recommended that while adding a DA, that the Parser be updated first, then the CAM; when deleting a DA, that the CAM be updated first, then the Parser. There is a small window when a DA is true in one database and not the other. This is not viewed as a problem.

The Parser database is partitioned into seven distinct sections. Some of these sections contain the list of enabled FCs, DAs etc. and some sections contain the forwarding information corresponding to individual user. These sections are described in detail as below.

### 7.2.1 FRAME CONTROL

The frame control section has been allocated one block of memory (Refer to Figure 18.) The FC is a one byte field which has 256 different possibilities. There is one entry (64-bit wide) for each possibility of FC and hence we have 256 such entries. The diagram below gives the details of each entry and a brief explanation. The logical relations and use of these bits can be better understood by following the Parsing Algorithm.



Note: Even if a packet qualifies for discard, a copy of such packet is sent to Promiscuous user, if it exists. If there is no Promiscuous user, the packet is then discarded.

BIT<63> : Discard = When this bit is "1", the packet containing that FC is discarded. The Discard bit for Nonrestricted, Restricted Tokens as well as for those FC values for which Frame Address bit indicates 16 bit address, should always remain set to "1".

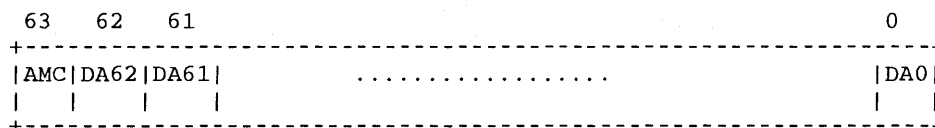
BIT<62:32> : Not Used By Parser. These bits are R/W by the Adapter Manager, but the Parser don't use them.

BIT<31:29> : FC Code = This is a three bit code assigned to 8 different classes of FCs. Some of these FC are defined today and some combinations have been reserved for future assignments. Please refer to DEMFA Firmware design Specification for more detailed definition of these bits.

BIT<28:0> : Not Used By Parser. These bits are R/W by the Adapter Manager, but the Parser don't use them.

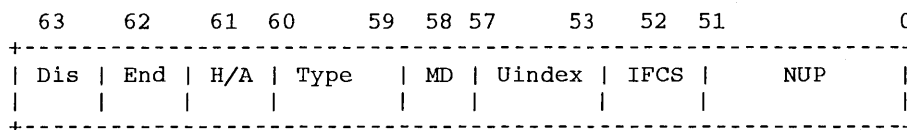
### 7.2.2 DESTINATION ADDRESSES

Destination Addresses (DA) have been assigned six blocks of the database memory. Each block corresponds to one byte of the six-byte long Destination Address. The values of the DA bytes are used as indices into these blocks. Now, for each byte, there are 256 possibilities(entries) and for each possibility there are 64 bit positions corresponding to 64 destination addresses. A zero in any of the 64 bit positions indicates that the byte used to index, is a part of a valid DA. If a zero is found in the same bit position in each of the six indexed quadwords, the DA is considered match. For example a "0" in 14th bit position of 56th entry in the first block means that that 14th DA has first byte as 56 (Decimal). Similarly, if a "0" is found in the 14th bit position of all remaining five blocks corresponding to remaining five bytes of DA, that DA is considered match. Each DA is initialized just once in this section. The 64th bit position has been assigned to the AMC user. This bit doesn't need any firmware modifications. This bit is never checked by the Parser Algorithm but it reserves a block of memory for the AMC User. The actual existence of an AMC User is checked via Control Register 1. Each entry of these six blocks of database, looks like as follows:



### 7.2.3 FILTERING DATA FOR FC AND DA ONLY FILTERING

This section has been assigned two blocks of memory for 512 different combinations of FC and DAs. Each entry in these blocks gives the final results of the filtering for packets whose filtering is based on FC and DA only. The combination of FC Code (3 bits) and DA\_INDEX (6 bits) is used to index these blocks of the database. The following diagram gives the details for each entry.



Note: Evenif a packet qualifies for discard, a copy of such packet is sent to Promiscuous user, if it exists. If there is no Promiscuous user, the packet is then discarded.

- BIT<63> : Discard = When this bit is "1", the packet containing that combination of FC and DA is discarded.
- BIT<62> : End = When this bit is "1", the packet does not need to go for further processing. It means that packet is filtered based on only FC and DA fields.
- BIT<61> : Host/Adapter Manager = If this bit is "1", the packet is forwarded to Host otherwise to the Adapter Manager.
- BIT<60:59> : Type = If the packet goes to the adapter manager, this type field shows the type of the packet. This 2-bit field defines four different types.

DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY

00 - XID/Test/Other  
01 - SMT/MAC  
10 - MOP  
11 - Error

Note : The Adapter Manager Firmware can use only "01" for FC and DA only filtering. The type "00" for FC and DA filtering is assigned by the Parser Internal Logic.

BIT<58> : Multiple Destination = If this bit is "1", the packet has multiple destinations. This bit is directly copied to the Forwarding vector. This bit doesn't include the Promiscuous user and/or Unknown User. The existence of a promiscuous user is known to the the Adapter Manager and the Port Driver and hence it is not indicated in the FV. The existence of Unknown user is indicated by Unk bit in the FV.

BIT<57:53> : User Index = This bit field gives the particular user index who should be receiving that packet. This bit field is directly copied to the Forwarding Vector.

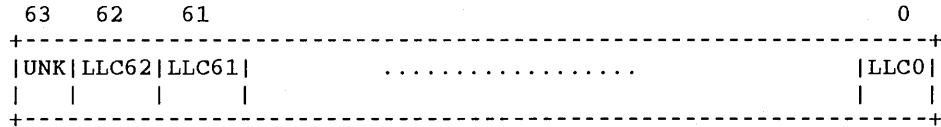
BIT<52> : Ignore FCS CRC Error = This bit is useful for only FC=Implementer and DA combination. When this bit is "1", the Parser to ignores the FCS CRC error for packet containing this combination of FC and DA. Evenif there is a error in this type of packet because of CRC being wrong, the packet is not marked as error packet and is forwarded.

BIT<51:0> : Not Used By Parser. These bits can be read "0" or "1" and writing to these bits have no impact on Parser functioning.

#### 7.2.4 DESTINATION SERVICE ACCESS POINT (DSAPs)

This section of memory is used for LLC filtering required for non SNAP SAP packets. The LLC filtering involves the filtering based either on DSAP or PID depending whether the packet is Non SNAP SAP or SNAP SAP.

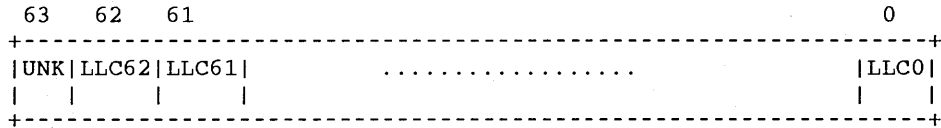
The DSAP section has been assigned one block of memory. The value of DSAP is used to index this block of memory. A zero in any bit position indicates that the DSAP has been enabled and DSAP has matched. The 64th bit position has been devoted to the Unknown user. Each DSAP is initialized just once in this entry. Similar to the AMC User discussion, the 64th bit in this section as well as in the next blocks of memory, is never used by the Parser Algorithm, but it reserves a block of memory for Unknown User. This bit also doesn't need any firmware modifications. The actual existence of an Unknown User is checked via Control Register 1. The entry looks as shown below.



**7.2.5 PROTOCOL IDENTIFICATIONS (PIDS)**

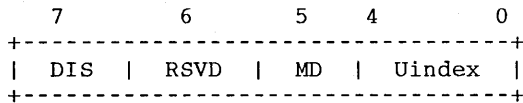
The LLC filtering based on PID is required for SNAP SAP packets. The description of PID section is similar to the description for destination addresses except that this section has five blocks of memory for each of the five bytes of PID. Also, the 64th bit position in each entry has been reserved for Unknown user similar as in case of DSAP. Each PID is initialized just once in this section. The entry also looks like the one in DSAP section.

The DSAP and PID are two mutually exclusive parts of filtering, means a packet is either filtered on DSAP or on PID. Also, the Parser implements 63 combinations LLCs (DSAPs and PIDs). **Hence the bit positions which are asserted here must be deasserted in DSAP entries and vice versa.**



**7.2.6 FILTERING DATA FOR FC,DA AND LLC FILTERING**

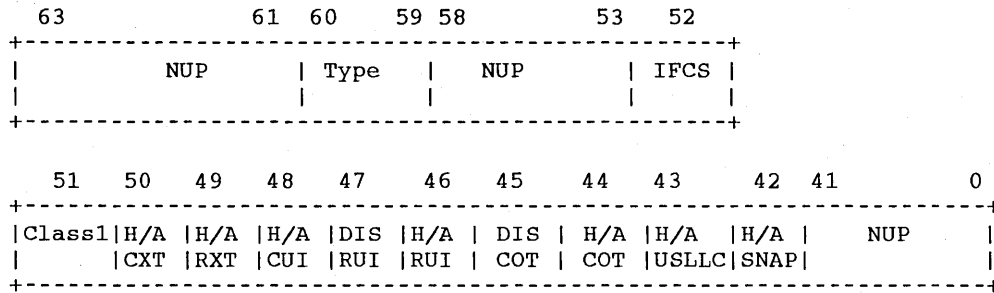
These sections have been assigned 16 blocks of memory means 32Kbytes of memory. The combination of FC Code(3bits), DA\_INDEX (6bits) and LLC\_INDEX (6bits), has been associated with one byte of information. The following picture depicts the contents of each byte.



- BIT<7> : Discard = When this bit is "1", the packet containing this combination of FC,DA and LLC is discarded.
- BIT<6> : Not Used By Parser. This bit can be read "0" or "1" and writing to this bit has no impact on Parser functioning.
- BIT<5> : Multiple Destination = If this bit is "1", the packet has multiple destinations. This bit doesn't include the Promiscuous user and/or Unknown User. The existence of a promiscuous user is known to the the Adapter Manager and the Port Driver and hence it is not indicated in the FV. The existence of Unknown user is indicated by Unk bit in the FV. This bit is directly copied to the Forwarding vector.
- BIT<4:0> : User Index = This gives the user index of the user who should be forwarded with this packet.

### 7.2.7 FILTERING DATA FOR A PARTICULAR USER FOR FC,DA AND LLC FILTERING

After the user index is received from the previous section, the Parsing is not completed until we have some more information about that user. This information is obtained from this section. This section has been assigned one block of memory, though only 32 entries corresponding to 32 users, are required. The Uindex is used to index this section of memory. Each entry contains the mode of LLC corresponding to that user and the destination of the packet which is based on each type of the packet. The following diagram depicts the details of each of these 32 entries.



BIT<63:61> : Not Used By Parser. These bits can be read "0" or "1" and writing to these bits have no impact on Parser functioning.

BIT<60:59> : Type = If the packet goes to the adapter manager, this type field shows the type of the packet. This 2-bit field defines four different types.

- 00 - XID/Test/Other
- 01 - SMT/MAC
- 10 - MOP
- 11 - Error

Note : The Adapter Manager Firmware can use only type "01" and "10" for FC,DA and LLC only filtering. Types "00" and "11" are assigned by the Parser Internal Logic.

BIT<58:53> : Not Used By Parser. These bits can be read "0" or "1" and writing to these bits have no impact on Parser functioning.

BIT<52> : Ignore FCS CRC Error =

When this bit is "1", the Parser ignores the FCS CRC error for packet containing this combination of FC,DA and LLC. Even if there is a error in this type of packet because of CRC being wrong, the packet is not marked as error packet and is forwarded.

BIT<51> : Class 1 = When this bit is "1", the user is supposed to be in class 1 LLC mode of operation, otherwise it is in UserSupplied LLC mode.

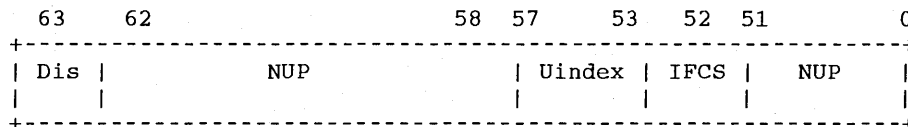
BIT<50> : Host/AM for Command XID/Test = If the packet is a command XID/Test and this bit is "1", the packet is forwarded to the Host, otherwise if this bit is "0", the packet is forwarded to Adapter Manager.



- BIT<49> : Host/AM for Response XID/Test = If the packet is a response XID/Test and this bit is "1", the packet is forwarded to the Host, otherwise if this bit is "0", the packet is forwarded to Adapter Manager.
- BIT<48> : Host/AM for Command UI = If the packet is a command UI and this bit is "1", the packet is forwarded to the Host, otherwise if this bit is "0", the packet is forwarded to Adapter Manager.
- BIT<47> : Discard for Response UI = If the packet is a response I and this bit is "1", the packet is discarded.
- BIT<46> : Host/AM for Response UI = If the packet is a response UI and this bit is "1", the packet is forwarded to the Host, otherwise if this bit is "0", the packet is forwarded to Adapter Manager.
- BIT<45> : Discard for Control\_Other = If the packet has a control field which is neither UI nor XID/Test and this bit is "1", the packet is discarded.
- BIT<44> : Host/AM for Control\_Other =  
 When this bit is "1", the packet is forwarded to the Host, otherwise the packet is forwarded to Adapter Manager.
- BIT<43> : Host/AM for User Supplied Mode = If the bit<51> is "0" and this bit is "1", the packet is forwarded to the Host, otherwise if this bit is "0", the packet is forwarded to Adapter Manager.
- BIT<42> : Host/AM for SNAP SAP packets = If the packet is a SNAP SAP, and this bit is "1", the packet is forwarded to the Host. If this bit is "0", the packet is forwarded to the AM.
- BIT<41:0> : Not Used By Parser.

## 7.2.8 FILTERING DATA FOR PROMISCUOUS USER(S)

The Promiscuous filtering is done after each discard condition. Please refer to Parsing Algorithm for various discard situations. In the case where the packet has been determined to go to a particular user(s), the promiscuous filtering is omitted. This is done because both the Adapter Manager and the Port Driver have a priori knowledge of existence of Promiscuous User(s) and they forward the packet to such user(s). There are eight entries corresponding to eight classes of FC, allocated to this section of filtering. These entries share the same block of memory as the previous section (Filter Data for a Particular User for FC,DA and LLC Filtering). The following diagram gives the details of these entries.



- BIT<63> : Discard = When this bit is "1", the packet is discarded. The "1" shows there is no promiscuous user.
- BIT<62:58> : Not Used By Parser.

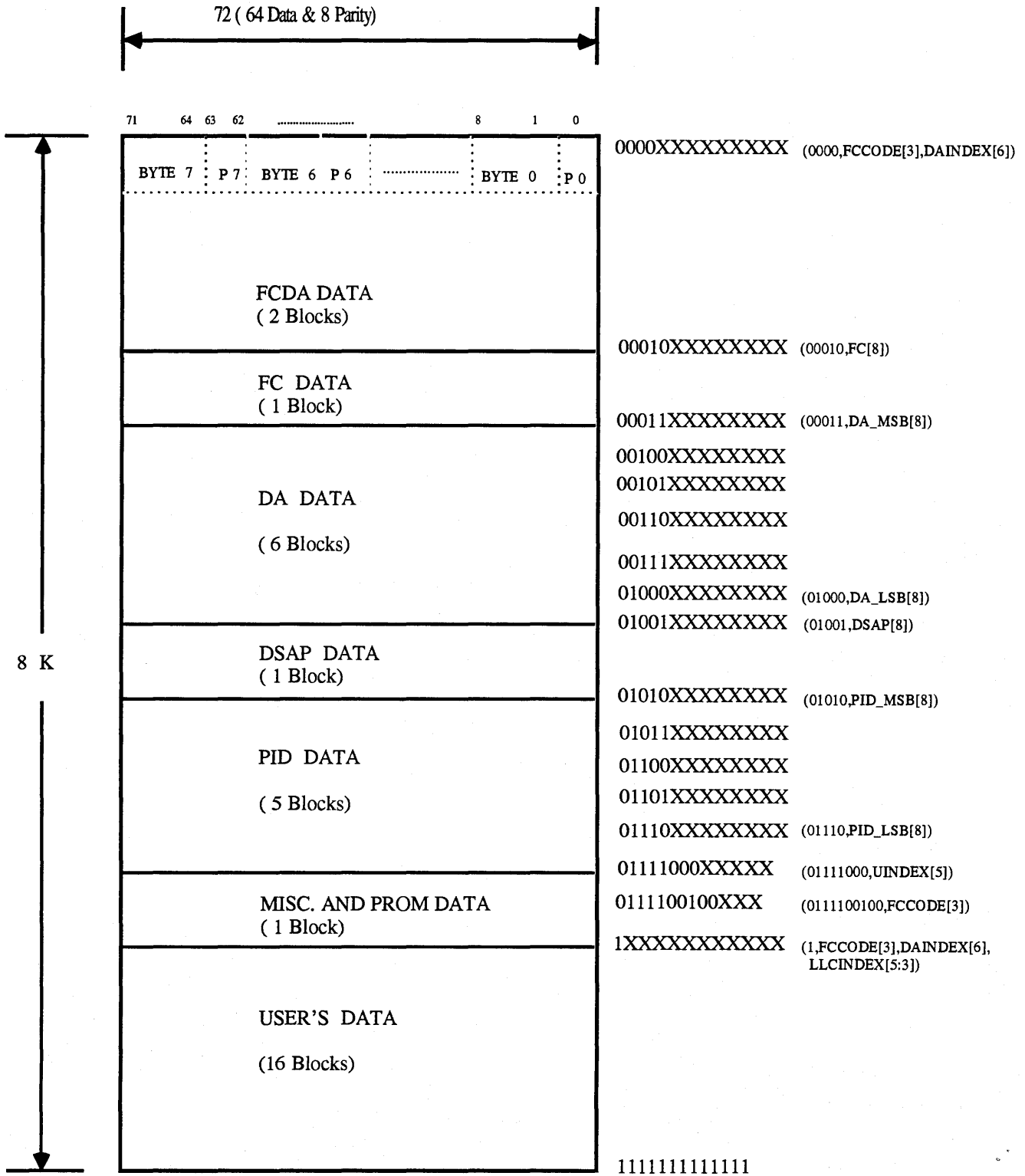
**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

BIT<57:53> : User Index = If the packet goes to the Host, this bit field gives the user index of the Promiscuous user who should be receiving that packet. This bit field is directly copied to the Forwarding vector.

BIT<52> : Ignore FCS CRC Error =  
When this bit is "1", the Parser ignores the FCS CRC error for the packet. Even if there is a error in this type of packet because of CRC being wrong, the packet is not marked as error packet and is forwarded.

BIT<51:0> : Not Used By Parser.

Figure 18: PARSER DATABASE MEMORY ALLOCATION



NOTE : EACH BLOCK IS DEFINED AS 256 X 72 .

# **CHAPTER 8**

## **FDDI REGISTERS**

For FDDI register definitions, refer to the RMC, MAC, and ELM specifications.

# CHAPTER 9

## DEMFA MISCELLANEOUS FEATURES

### 9.1 DEMFA MODULE DATA INTEGRITY

Data integrity is of the essence and extensive efforts have been made to protect data from corruption as it passes through the DEMFA. As can be seen from Figure 19 the main data stream between the FDDI RING and the XMI is protected by, starting from the FDDI end, CRC. Within the MAC, CRC is overlapped with byte parity. Byte parity is maintained through the packet buffer. At the ESP interface of the PMC, byte parity is exchanged for longword parity, which is the data protection of the XMI bus.

Within the 68020 subsystem, the following is the rule: the 68020 does not support any form of data protection. However, the local memory, S\_RAM, is protected by byte parity which the AMI gate array generates/checks. The E\_E (EEPROM) in which the test and functional codes reside is protected by checksum, as is the DPA ROM.

The PMC memory sub-systems offer the following protection. All packet data is protected by byte parity, as are buffer descriptors. The page table entries are protected as follows: the PPN (Physical Page Number) is 15 bits + 1 bit parity, the FV (Forwarding Vector) is 15 bits + 1 bit parity and the O/C (Ownership and Color) is 3 bits + 1 bit parity.

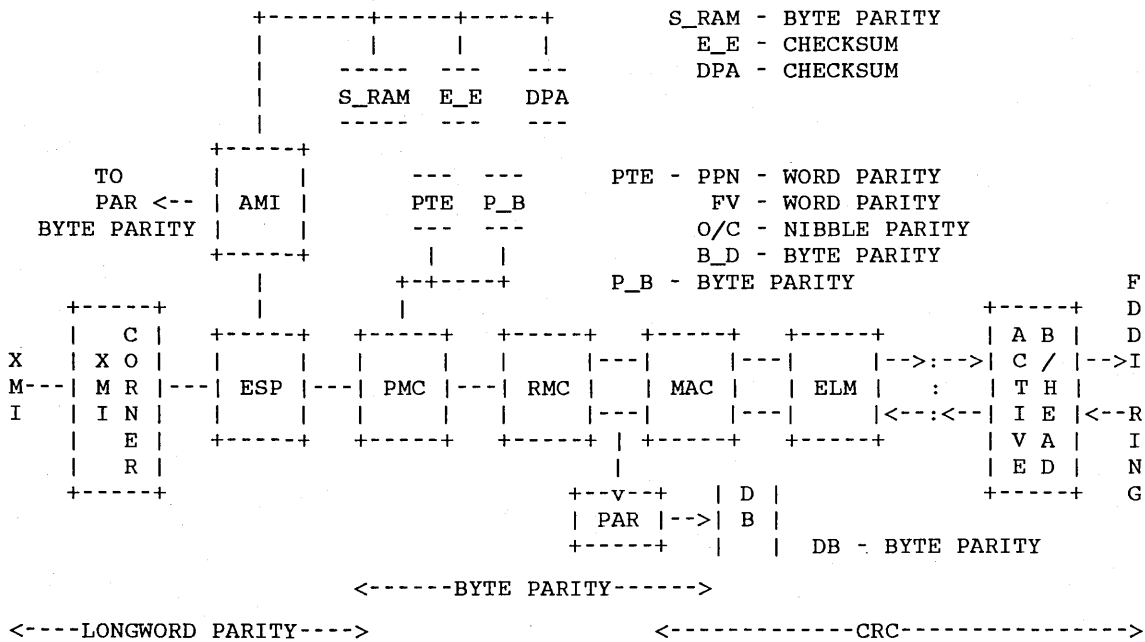
If the SRAM parity error is a 'soft' single bit error, which is the most likely, then initializing the adapter with Node Reset will recover the adapter to a fully working state and can therefore have its EEPROM updated. If the SRAM parity error is hard, then initializing the adapter with a Node Reset will result in self test reporting another SRAM parity error. Under these circumstances the module is physically broken and requires repair. EEPROM update cannot take place because we specifically stop the 68020 from working when we detect an SRAM parity error. Once the broken RAM is repaired, the EEPROM should be able to be updated by the normal means.

The PAR (Parser) is connected to the 68020 subsystem with the NP bus. This bus is protected with the byte parity. Also, the PAR has associated DB (Database) which is protected by the byte parity.

### 9.2 FDDI MAINTENANCE SERVICES

The DEMFA implements FDDI maintenance operations as specified in the FDDI Maintenance Operations Functional Specification. The description and message formats of these operations can be found in that document.

Figure 19: THE DEMFA DATA INTEGRITY SCHEME



The DEMFA supports the following set of maintenance functions, for both Ethernet and 802 packets.

- Loop Test. The DEMFA decodes Loopback messages and either forwards them to another node or delivers them to the port driver, depending on the content of the message.
- EEPROM Update. The EEPROM can be updated locally and remotely with local host help. All revision levels of hardware and firmware are located in the EEPROM. Updates are described in detail in the DEMFA Firmware Functional specification.
- Remote Console. The DEMFA processes the following types of MOP messages:
  - Request ID. The DEMFA sends a System ID message to the requesting node in response to a Request ID message.
  - System ID. The DEMFA sends System ID messages on a regular basis to the Remote Console Server multicast address. It alternates 802 and Ethernet format messages every 4-6 minutes, so that each format is repeated every 8-12 minutes.
  - Request Counters. The DEMFA sends a Counters message to the requesting node in response to a Request Counters message.
  - Boot. The DEMFA causes a bus reset upon receipt of a valid Boot message.
  - Console Carrier. The console carrier commands will be forwarded to the Host whenever Host User for the frames is defined in the USTART command.

### 9.3 INTERLOCKED OPERATION

The DEMFA does not implement any interlocked instructions. However, if an interlocked read or write is directed to the DEMFA, it will treat the instruction as a regular read or write.

## 9.4 SELF TEST PROCEDURE

Self test is run on power up or node reset (setting of the *Node Reset (NRST)* bit in the XMI Bus Error Register). It takes less than 10 seconds.

Self test consists of the following parts:

1. Execute self test, writing results into the Power Up Diagnostic Register.
2. Clear port registers, thus indicating the port state *Resetting*.
3. Write the XMI Device Register with device type and revision level.
4. Clear *Self Test Fail* bit in the XMI Bus Error Register.
5. Jump to operational firmware.

Note that self test includes the LAN Address ROM Test described in the DNA NI Node Product Architecture Specification.

## 9.5 STATUS LEDS

The DEMFA Module has two LEDS (Yellow and Green). The combined states of these LEDS indicate the Field Replaceable Units (FRUs) failure. The following table describe the various combinations of these LEDS.

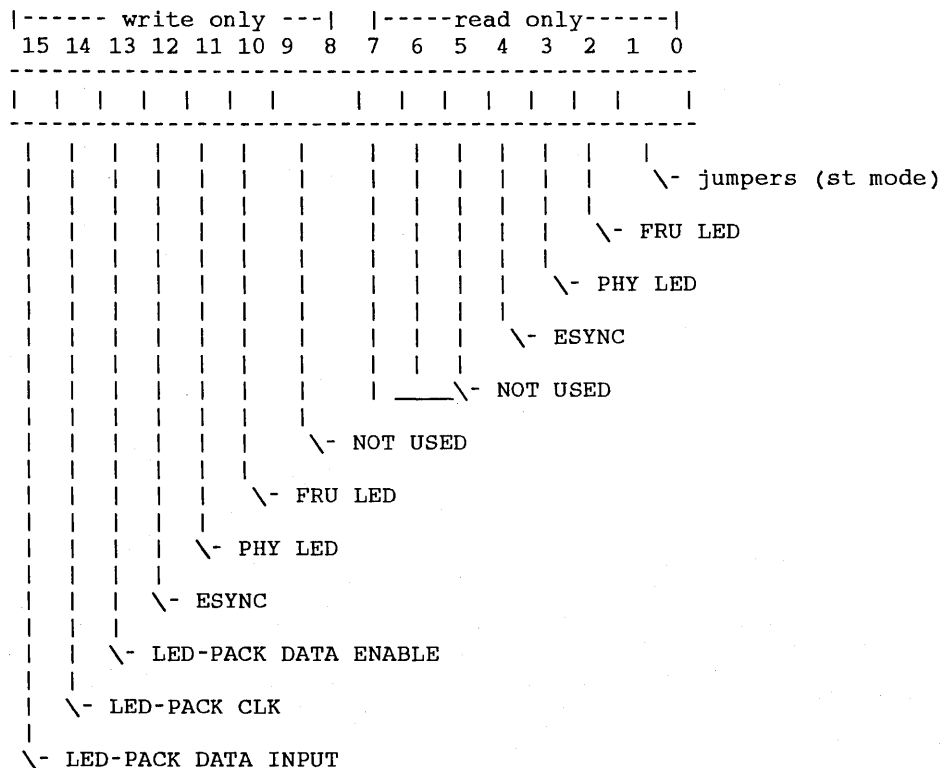
**Table 7: DEMFA MODULE STATUS LEDS**

| Self-Test LED (Yellow) | Status LED (Green) | Description                                                                                                                                                                                                                           |
|------------------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OFF                    | OFF                | Self-Test Failed - The DEMFA Module is faulty                                                                                                                                                                                         |
| OFF                    | ON                 | Self-Test Failed - The Active Bulkhead or Adapter Cable is faulty                                                                                                                                                                     |
| ON                     | BLINKING           | The ESP Special Test is not operational. The Possible causes are :<br>* Host Driver not installed or not properly enabled<br>* Host Driver and Firmware are not compatible<br>* XMI backplane or XMI interface not operating properly |
| ON                     | OFF                | The ESP Special Test Failed - DEMFA Module is faulty                                                                                                                                                                                  |
| ON                     | ON                 | The DEMFA Module has passed the Self-Test and the ESP Special test                                                                                                                                                                    |

The External Diagnostic CSR register is shown on the following page:

Figure 20: EXTERNAL DIAGNOSTIC CSR REGISTER

Diagnostic CSR register: Structural Overview & Description  
 gbl\$gw\_diag\_csr



```

gbl$gw_diag_csr      EQU      $100700 * Diagnostic CSR

Bitdef DCSR,DATA_IN,15 * LED-pack Data In; 0 = LED off, 1=on
Bitdef DCSR,CLK,14 * LED-pack Clock; 0 to 1 clocks
Bitdef DCSR,DATA_EN,13 * Data Enable, 0 = disable, 1 = enable
Bitdef DCSR,ESYNC_W,12 * ESYNC Write; 1 = error, 0 = no error
Bitdef DCSR,PHY_LED_W,11 * PHY LED; 1 = LED on, 0 = LED off
Bitdef DCSR,FRU_LED_W,10 * FRU LED; 1 = LED on, 0 = LED off
Flddef DCSR,DCSR_UNUSED2,8,2 * Not used
Flddef DCSR,DCSR_UNUSED,5,3 * Not used
Bitdef DCSR,ESYNC_R,4 * ESYNC Read
Bitdef DCSR,PHY_LED_R,3 * PHY LED Read
Bitdef DCSR,FRU_LED_R,2 * FRU LED Read
Bitdef DCSR,W2,1 * Jumper W2
Bitdef DCSR,W1,0 * Jumper W1
  
```

```

Self-test Mode = W2-W1
11 = Normal Mode
10 = MFG #1 - Execute mfg once
    and start firmware.
01 = MFG #2 - Execute mfg contin.
    and don't start firmware.
00 = MFG #3 - Execute mfg tests
    with bulkhead/ext loopback.
  
```



# CHAPTER 10

## DEMFA MODULE SPECIFICATION

### 10.1 DEMFA MODULE SPECIFICATION

#### 10.1.1 OVERVIEW

DEMFA consists of the XMI module, active bulkhead and a cable which connects XMI module and active bulkhead. This chapter will specify module power dissipation, clock, mechanical dimensions and MTBF.

#### 10.1.2 POWER DISSIPATION

##### 10.1.2.1 DEMFA XMI MODULE

**Table 8: ELECTRICAL CHARACTERISTICS**

| Voltage | Current (amps) | Power Dissipation (watts) |
|---------|----------------|---------------------------|
| +5V     | 12.4           | 62                        |

##### 10.1.2.2 DEMFA ACTIVE BULKHEAD MODULE

**Table 9: ELECTRICAL CHARACTERISTICS**

| Voltage | Current (amps) | Power Dissipation (watts) |
|---------|----------------|---------------------------|
| +5.0V   | 0.09           | 0.45                      |
| -5.2V   | 0.98           | 5.1                       |

#### 10.1.3 DEMFA CLOCK

There are two primary clocks on DEMFA module, 64 ns cycle clock and 80 ns cycle clock. The 64 ns clock is derived from XMI corner. The 64 ns clock is available in two different phase relationships, CLK34 and CLK61. The 80 ns clock (BYTCLK) is originated from 50 Mhz oscillator on DEMFA module. Another secondary clock 40 ns clock (SYMCLK) is derived from the 80 ns clock. The AMI, PMC and ESP gate array use CLK34, CLK61 and BYTCLK. The Parser gate array and FDDI chipsets use BYTCLK and SYMCLK. The synchronization between the two primary clocks is done in PMC gate array. The DEMFA module also supplies SYMCLK to active bulkhead through the 15-foot cable.

## 10.1.4 MECHANICAL DIMENSIONS

### 10.1.4.1 DEMFA XMI MODULE

An XMI module is a board that is 9.18" high, 11.024" wide and 0.093"(typ.) thick. The DEMFA module is a 10 layer module with 6 signal layers, 2 power planes and 2 ground planes. The overall physical dimensions of an XMI module, in inches, are

**Table 10: XMI MODULE DIMENSIONS**

| Dimension                                | Minimum | Maximum |
|------------------------------------------|---------|---------|
| Height<br>(top to bottom)                | 9.175   | 9.187   |
| Width<br>(front to connector)            | 11.019  | 11.031  |
| Thickness<br>(within 0.65" of connector) | 0.083   | 0.103   |
| Component projection<br>(on side 1)      | 0       | 0.420   |
| (on side 2)                              | 0       | 0.182   |

### 10.1.4.2 DEMFA ACTIVE BULKHEAD

The mechanical dimensions of active bulkhead are 4" high, 2.6" wide and 0.057" thick(not include solder mask thickness). The active bulkhead is a 4 layers modules(2 signal layers, 1 power plane and 1 ground plane).

## 10.1.5 OPERATING ENVIRONMENT

The DEMFA option is required to operate within the limits of DEC Standard 102(Rev. D) for a class B enviroment. The following table gives more detailed information of the operating environment.

**Table 11: OPERATING ENVIRONMENTAL REQUIREMENTS**

| Parameter                     | Value                                                                                                       |
|-------------------------------|-------------------------------------------------------------------------------------------------------------|
| Temperature (at sea level)    | 10 degree C to 40 degree C (50 degree F to 104 degree F)                                                    |
| Temperature (above sea level) | Reduce the maximum operating ambient temperature by 1.8 degree C per 1000 meters (1 degree F per 1000 feet) |
| Maximum rate of change        | 11 degree C per hour +/- 2 degree C per hour (20 degree F per hour +/- 4 degree F per hour)                 |
| Relative Humidity             | 10% to 90%                                                                                                  |
| Wet-Bulb Temperature          | 28 degree C (82 degree F)                                                                                   |
| Altitude                      | 2.4 KM (8000 feet)                                                                                          |

### 10.1.6 NON-OPERATING ENVIRONMENT

The following table lists storage conditions for the DEMFA module.

**Table 12: NON-OPERATING ENVIRONMENTAL REQUIREMENTS**

| Parameter         | Value                                                      |
|-------------------|------------------------------------------------------------|
| Temperature       | -40 degree C to 66 degree C (-40 degree F to 151 degree F) |
| Relative Humidity | up to 95% (non-condensing)                                 |
| Altitude          | 4.9 km (16,000 ft)                                         |

### 10.1.7 MEANTIME BETWEEN FAILURES

The DEMFA option has the following MeanTime Between Failures(MTBF). These figures were calculated using Predic.

**Table 13: DEMFA MTBF (HOURS)**

| Operating Temperature (C) | Option | Module | Bulkhead |
|---------------------------|--------|--------|----------|
| 25                        | 172.5K | 214.0K | 900K     |
| 40                        | 78.9k  | 86.7k  | 860k     |
| 50                        | 42.6k  | 45.0k  | 830k     |

## APPENDIX A

### DEMFA MODULE I/O PINNING

The following pages show a table of the DEMFA I/O signals.

#### A.1 XMI INTERFACE SIGNALS

Signal names enclosed in ( ) appear on the XMI backplane, but are not used on the DEMFA module.

| PIN | NAME        | PIN | NAME                   |
|-----|-------------|-----|------------------------|
| A01 | XMI D<9> L  | A31 | XMI SUP L              |
| A02 | XMI D<8> L  | A32 | GROUND                 |
| A03 | +5VBB       | A33 | +5VBB                  |
| A04 | XMI D<7> L  | A34 | +5VBB                  |
| A05 | XMI D<5> L  | A35 | NC ( XMI SPAREB2 L )   |
| A06 | XMI D<4> L  | A36 | GROUND                 |
| A07 | XMI D<2> L  | A37 | NC ( XMI SPAREB3 L )   |
| A08 | XMI D<6> L  | A38 | XMI D<18> L            |
| A09 | XMI D<3> L  | A39 | NC ( XMI SPAREB4 L )   |
| A10 | XMI FAULT L | A40 | GROUND                 |
| A11 | XMI D<29> L | A41 | XMI D<0> L             |
| A12 | XMI P<0> L  | A42 | NC ( XMI TOYBBUPWR H ) |
| A13 | GROUND      | A43 | GROUND                 |
| A14 | XMI D<31> L | A44 | GROUND                 |
| A15 | XMI D<28> L | A45 | XMI D<30> L            |
| A16 | XMI D<13> L | A46 | NC ( XMI TOYBBU OK L ) |
| A17 | XMI D<14> L | A47 | XMI D<10> L            |
| A18 | +5VBB       | A48 | +5VBB                  |
| A19 | +5VBB       | A49 | NC ( XMI CON RCV H )   |

DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
 14-March-1991 - INTERNAL USE ONLY

| PIN | NAME        | PIN | NAME                    |
|-----|-------------|-----|-------------------------|
| A20 | XMI D<19> L | A50 | XMI D<12> L             |
| A21 | XMI D<20> L | A51 | XMI D<1> L              |
| A22 | XMI D<21> L | A52 | NC ( XMI CON XMIT H )   |
| A23 | XMI D<17> L | A53 | XMI D<16> L             |
| A24 | XMI D<15> L | A54 | NC ( XMI RUN L )        |
| A25 | XMI D<32> L | A55 | XMI RESET L             |
| A26 | XMI D<33> L | A56 | GROUND                  |
| A27 | XMI D<34> L | A57 | XMI D<11> L             |
| A28 | GROUND      | A58 | GROUND                  |
| A29 | GROUND      | A59 | XMI D<36> L             |
| A30 | XMI D<35> L | A60 | NC ( XMI CON SECURE H ) |
| B01 | XMI D<25> L | B31 | XMI D<40> L             |
| B02 | XMI D<24> L | B32 | GROUND                  |
| B03 | +5V         | B33 | +5V                     |
| B04 | XMI D<23> L | B34 | +5V                     |
| B05 | XMI D<22> L | B35 | XMI D<26> L             |
| B06 | XMI D<53> L | B36 | GROUND                  |
| B07 | XMI D<52> L | B37 | XMI BAD L               |
| B08 | XMI D<51> L | B38 | XMI D<54> L             |
| B09 | XMI D<50> L | B39 | GROUND                  |
| B10 | XMI D<49> L | B40 | NC ( XMI BOOT L )       |
| B11 | XMI D<48> L | B41 | XMI UPDATE EN H         |
| B12 | XMI D<47> L | B42 | GROUND                  |
| B13 | +5V         | B43 | -5.2V                   |
| B14 | XMI D<46> L | B44 | -5.2V                   |
| B15 | XMI D<45> L | B45 | XMI D<44> L             |
| B16 | XMI D<38> L | B46 | XMI D<27> L             |
| B17 | XMI D<41> L | B47 | NC ( XMI SPARE0 L )     |
| B18 | +5V         | B48 | +5V                     |
| B19 | +5V         | B49 | XMI D<42> L             |
| B20 | XMI D<39> L | B50 | GROUND                  |
| B21 | D<58> L     | B51 | GROUND                  |
| B22 | XMI D<55> L | B52 | GROUND                  |
| B23 | XMI D<56> L | B53 | XMI D<37> L             |
| B24 | XMI D<57> L | B54 | GROUND                  |
| B25 | XMI D<59> L | B55 | XMI P<1> L              |
| B26 | XMI D<61> L | B56 | GROUND                  |
| B27 | XMI D<62> L | B57 | XMI D<60> L             |
| B28 | +5V         | B58 | -5.2V                   |
| B29 | +5V         | B59 | XMI LOCKOUT             |
| B30 | XMI D<63> L | B60 | GROUND                  |
| C01 | GROUND      | C31 | XMI D<43> L             |
| C02 | XMI ID<5> L | C32 | GROUND                  |

DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
 14-March-1991 - INTERNAL USE ONLY

| PIN | NAME            | PIN | NAME                  |
|-----|-----------------|-----|-----------------------|
| C03 | NC (+3V)        | C33 | -2V                   |
| C04 | GROUND          | C34 | -2V                   |
| C05 | XMI HOLD L      | C35 | XMI CNF<2> L          |
| C06 | XMI NOD ID<0> H | C36 | GROUND                |
| C07 | XMI NOD ID<1> H | C37 | +12V                  |
| C08 | XMI NOD ID<2> H | C38 | XMI CMD REQ L         |
| C09 | XMI NOD ID<3> H | C39 | -12V                  |
| C10 | GROUND          | C40 | XMI RES REQ L         |
| C11 | XMI F<1> L      | C41 | GROUND                |
| C12 | XMI F<0> L      | C42 | XMI GRANT L           |
| C13 | GROUND          | C43 | GROUND                |
| C14 | GROUND          | C44 | GROUND                |
| C15 | GROUND          | C45 | XMI PHASE L           |
| C16 | XMI AC LO L     | C46 | NC ( XMI SPAREBT0 L ) |
| C17 | XMI P<2> L      | C47 | XMI CNF<0> L          |
| C18 | NC ( +3V )      | C48 | -2V                   |
| C19 | NC ( +3V )      | C49 | XMI CNF<1> L          |
| C20 | XMI ID<4> L     | C50 | NC ( XMI SPAREBT1 L ) |
| C21 | XMI ID<3> L     | C51 | GROUND                |
| C22 | XMI ID<2> L     | C52 | GROUND                |
| C23 | XMI ID<1> L     | C53 | GROUND                |
| C24 | XMI ID<0> L     | C54 | XMI ERR DEF L         |
| C25 | XMI F<3> L      | C55 | XMI DC LO L           |
| C26 | XMI F<2> L      | C56 | GROUND                |
| C27 | GROUND          | C57 | XMI TIME L            |
| C28 | GROUND          | C58 | GROUND                |
| C29 | GROUND          | C59 | XMI TIME H            |
| C30 | GROUND          | C60 | GROUND                |

## A.2 DEMFA ACTIVE BULKHEAD INTERFACE SIGNALS

| PIN | NAME             | PIN | NAME          |
|-----|------------------|-----|---------------|
| D01 | VCC              | D31 | GND           |
| D02 | XMTDATA<4>       | D32 | MINUS 5.2V    |
| D03 | ABULKHD LED ON L | D33 | MINUS 5.2V    |
| D04 | XMTDATA<1>       | D34 | CDCLOOPBACK L |
| D05 | XMITDATA<2>      | D35 | GND           |
| D06 | GND              | D36 | RCVDATA<2>    |
| D07 | XMTDATA<0>       | D37 | RCVDATA<3>    |
| D08 | XMTDATA<3>       | D38 | GND           |
| D09 | GND              | D39 | RCVDATA<4>    |
| D10 | SYMBOLCLK        | D40 | RCVDATA<0>    |
| D11 | FOXMTOFF L       | D41 | GND           |
| D12 | GND              | D42 | CDCSYMBOLCLK  |
| D13 | MINUS 5.2V       | D43 | RCVDATA<1>    |
| D14 | MINUS 5.2V       | D44 | SIG DET       |
| D15 | GND              | D45 | VCC           |
| D16 | RSVD             | D46 | RSVD          |
| D17 | RSVD             | D47 | RSVD          |
| D18 | RSVD             | D48 | RSVD          |
| D19 | RSVD             | D49 | RSVD          |
| D20 | RSVD             | D50 | RSVD          |
| D21 | RSVD             | D51 | RSVD          |
| D22 | RSVD             | D52 | RSVD          |
| D23 | RSVD             | D53 | RSVD          |
| D24 | RSVD             | D54 | RSVD          |
| D25 | RSVD             | D55 | RSVD          |
| D26 | RSVD             | D56 | RSVD          |
| D27 | RSVD             | D57 | RSVD          |
| D28 | RSVD             | D58 | RSVD          |
| D29 | RSVD             | D59 | RSVD          |
| D30 | RSVD             | D60 | RSVD          |
| E01 | RSVD             | E31 | RSVD          |
| E02 | RSVD             | E32 | RSVD          |
| E03 | RSVD             | E33 | RSVD          |
| E04 | RSVD             | E34 | RSVD          |
| E05 | RSVD             | E35 | RSVD          |
| E06 | RSVD             | E36 | RSVD          |
| E07 | RSVD             | E37 | RSVD          |

**NOTE:**

VCC = +5.0V

DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION  
14-March-1991 - INTERNAL USE ONLY

| PIN | NAME             | PIN | NAME      |
|-----|------------------|-----|-----------|
| E08 | RSVD             | E38 | RSVD      |
| E09 | RSVD             | E39 | RSVD      |
| E10 | RSVD             | E40 | RSVD      |
| E11 | RSVD             | E41 | RSVD      |
| E12 | RSVD             | E42 | RSVD      |
| E13 | RSVD             | E43 | RSVD      |
| E14 | RSVD             | E44 | RSVD      |
| E15 | RSVD             | E45 | RSVD      |
| E16 | GND              | E46 | GND       |
| E17 | GND              | E47 | RSVD      |
| E18 | RSVD             | E48 | RSVD      |
| E19 | VCC              | E49 | VCC       |
| E20 | VCC              | E50 | VCC       |
| E21 | RSVD             | E51 | RSVD      |
| E22 | GND              | E52 | GND       |
| E23 | DATA IN          | E53 | GND       |
| E24 | CLK              | E54 | GND       |
| E25 | DATA EN          | E55 | GND       |
| E26 | ESYNC            | E56 | GND       |
| E27 | ABULKHD LED ON L | E57 | GND       |
| E28 | GND              | E58 | GND       |
| E29 | GND              | E59 | Npdata<1> |
| E30 | GND              | E60 | Npdata<0> |

**NOTE:**

VCC = +5.0V



## **APPENDIX B**

### **GATE ARRAY PINNING**

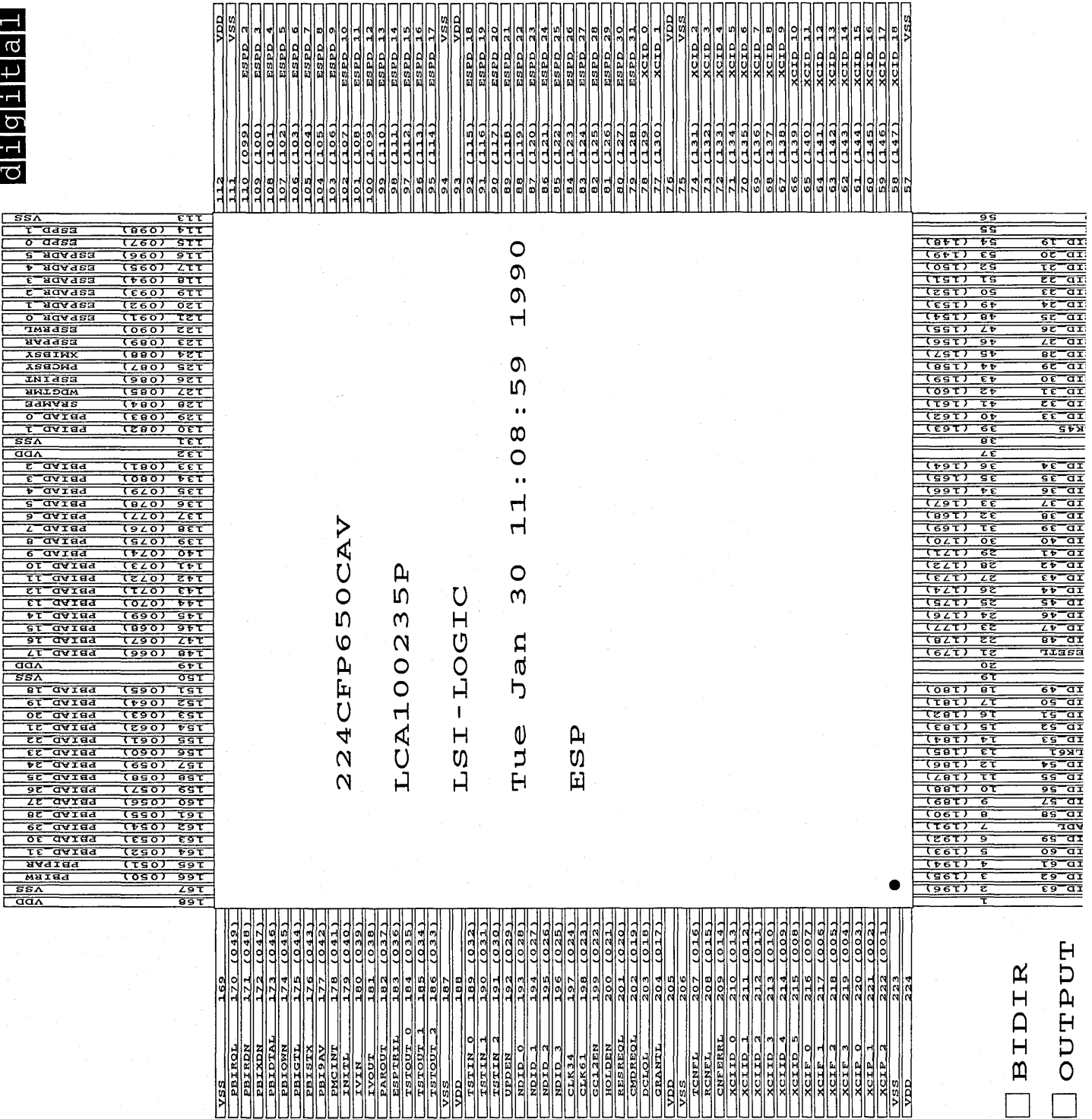
#### **B.1 ESP GATE ARRAY PINNING**

The following diagram (Figure 21) gives the pinout of the ESP Gate Array.

Figure 21: ESP GATE ARRAY PIN DIAGRAM

digital

TOP VIEW



□ BIDIR  
 □ OUTPUT

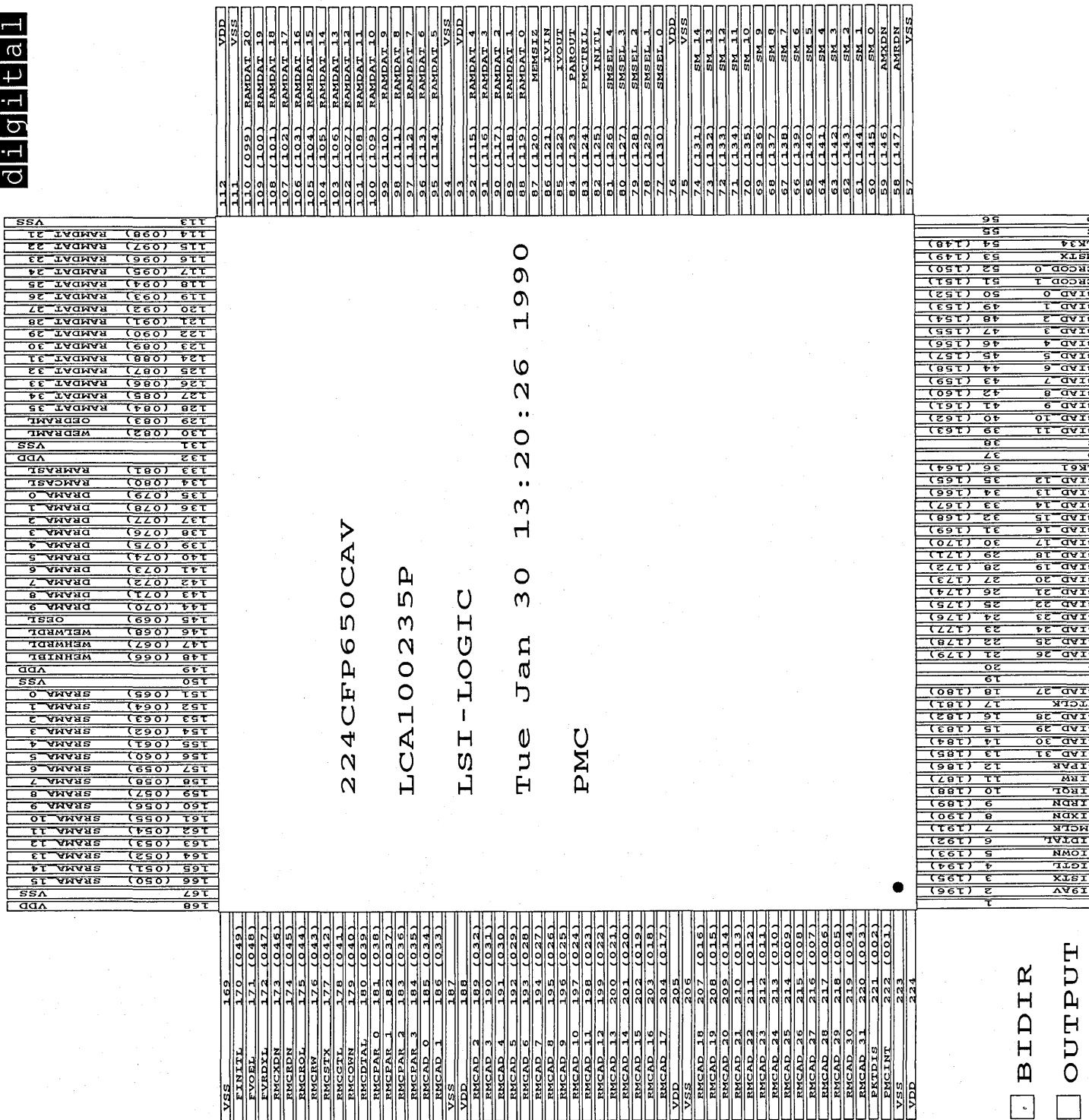
## B.2 PMC GATE ARRAY PINNING

The following diagram (Figure 22) gives the pinout of the PMC Gate Array.

Figure 22: PMC GATE ARRAY PIN DIAGRAM

digital

TOP VIEW



### B.3 AMI GATE ARRAY PINNING

The following diagram (Figure 23) gives the pinout of the AMI Gate Array.



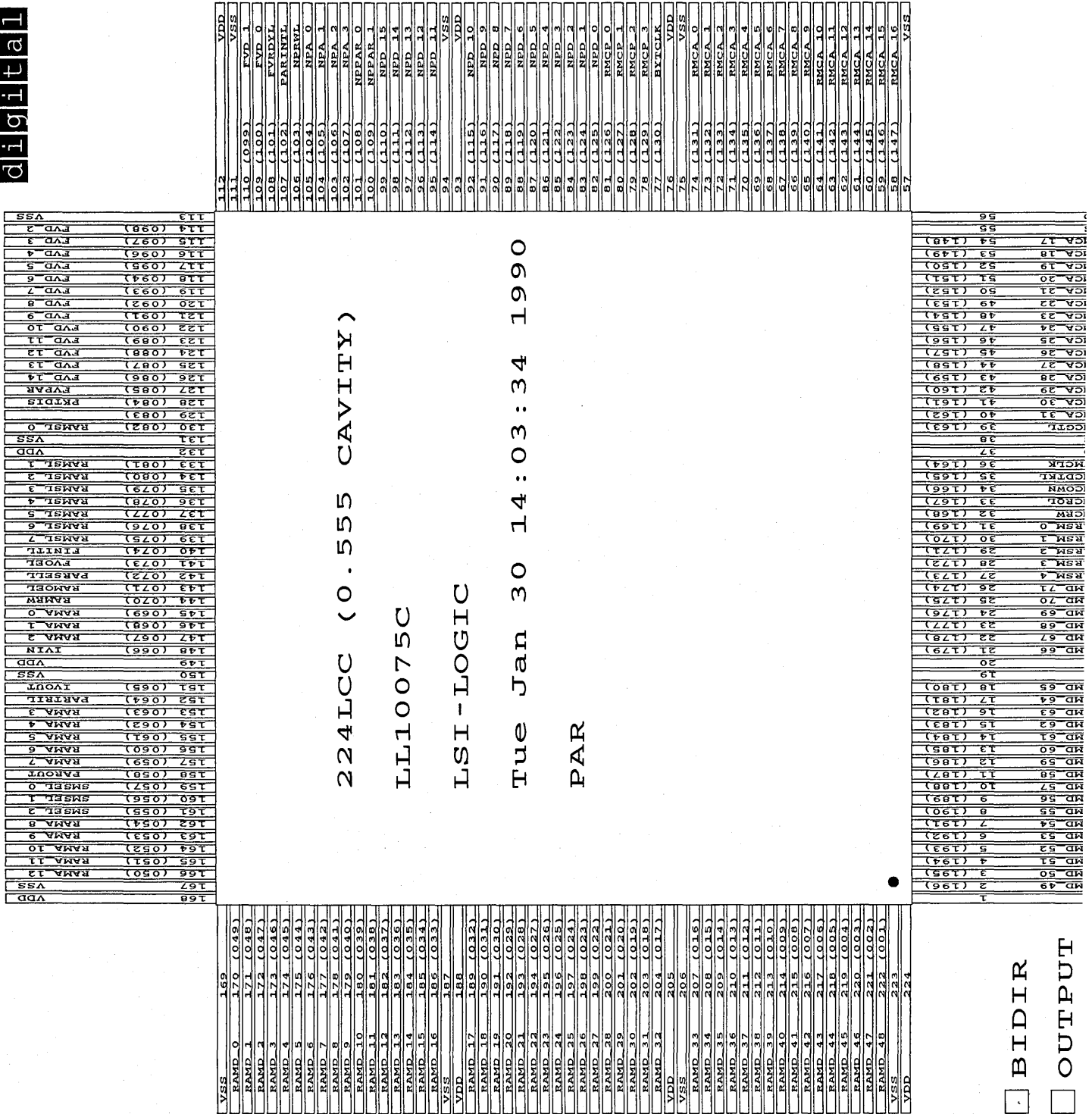
## B.4 PAR GATE ARRAY PINNING

The following diagram (Figure 24) gives the pinout of the Parser Gate Array.

Figure 24: PARSER GATE ARRAY PIN DIAGRAM

digital

TOP VIEW



BIDIR  
 OUTPUT



**APPENDIX C**  
**ACTIVE BULKHEAD**

DEMFA ACTIVE BULKHEAD

MODULE

FUNCTIONAL SPECIFICATION

(54-18847-01)

---

| REVISION HISTORY |           |               |              |
|------------------|-----------|---------------|--------------|
| REVISION         | DATE      | AUTHOR        | NOTES        |
| 0.0              | 09-DEC-88 | Larry DeRenne | First Draft  |
| 1.0              | 03-APR-89 | Larry DeRenne | Second Draft |
| 2.0              | 25-AUG-89 | Larry DeRenne | Third Draft  |
| 3.0              | 25-NOV-90 | Larry DeRenne | Four Draft   |
| 4.0              | 21-MAR-90 | Larry DeRenne | Fifth Draft  |

---

**AUTHOR:**  
Larry DeRenne  
LTN2-2/C08  
21-MAR-90  
226-6711

## C.1 INTRODUCTION

The Active Bulkhead is the FDDI, (Fiber Connection), end of the DEMFA Adaptor.

## C.2 RELATED DOCUMENTS

DEMFA Hardware Functional Specification

DEMFA Hardware Functional Specification

CDCT Specification

CDCR Specification

FOR Specification

FOT Specification

FDDI Specification

## C.3 PRODUCT DESCRIPTION

The Active Bulkhead is a 4-layer printed circuit board with intrusive-leaded and surface-mounted components. The circuit board is a standard size of (2.6 x 4.0 x .062). Components are mounted on side one only. The circuit board is mounted to a conduction plate. This assembly can be attached to the I/O bulk-head of a system using a dual, quad (default), or octal panel to provide the fiber connection to FDDI.

The CDCT and CDCR are kept at a proper operating temperature by use of a heat sink. The proper operating temperature is 85 Deg C for the junction temperatures. The heat sink, or (conduction plate) is bonded to the CDC'S with reparable thermal epoxy. The conduction plate is riveted to either a dual, quad, or octal panel.

The Active Bulkhead module talks to the main module residing in the system backplane, via a 30 conductor cable. There are two five bit parallel groups of data travelling along the cable. One to the main module from the Active Bulkhead and one to the Active Bulkhead from the main module. The data travels at the rate of 25 Mhz.

Power is also supplied to the Active Bulkhead Module by the main module, via the 30 conductor cable.

A green LED which is visible from outside of the cabinet indicates the following conditions.

|          |                                 |
|----------|---------------------------------|
| OFF      | Management Disable              |
| ON-BLINK | Connecting - not yet successful |
| ON-SOLID | Good PHY Connection             |

### NOTE

**For the PHY LED control bit, refer to the External Diagnostic CSR Register at the end of the DEMFA Miscellaneous Features section.**

**DIGITAL EQUIPMENT CORPORATION - DEMFA HARDWARE FUNCTIONAL SPECIFICATION**  
**14-March-1991 - INTERNAL USE ONLY**

**C.3.1 PARTSLIST**

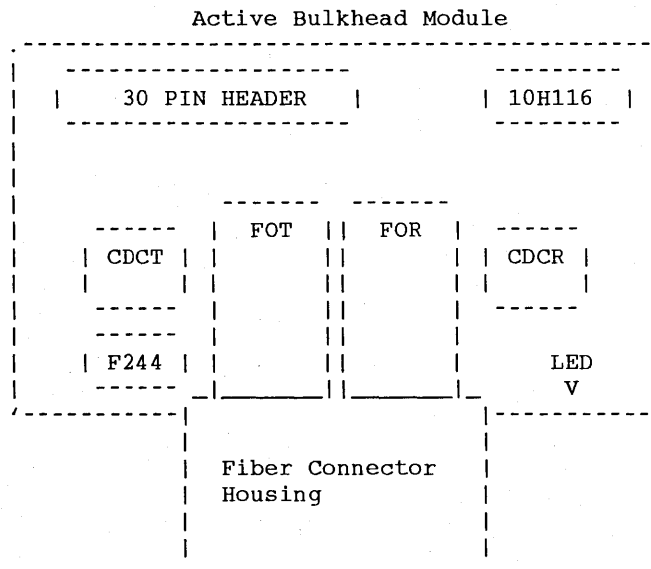
| ITEM | QTY | DESCRIPTION                                                         | QUAL | DEC PART NO.  | COST/EA. |
|------|-----|---------------------------------------------------------------------|------|---------------|----------|
| 1.   | 1   | 4-layer PC board raw etch (.062)<br>2.60 X 4.00 (STANDARD SIZE,FR4) | NO   | (50-18846-01) | 35.000   |
| 2.   | 1   | Clock and data conversion rcvr<br>J lead (surf mount) [sb730]       | NO   | (15-29846-02) | 5.000    |
| 3.   | 1   | Clock and data conversion xmit<br>J lead (surf mount) [sb729]       | NO   | (15-29847-02) | 5.000    |
| 4.   | 1   | Sumitimo fiber optic (xmitter)                                      | NO   | (24-32322-01) | 425.000  |
| 5.   | 1   | Sumitomo fiber optic (receiver)                                     | NO   | (24-32323-01) | 425.000  |
| 6.   | 1   | Fiber optic connector housing                                       | NO   | (12-31200-04) | 13.000   |
| 7.   | 1   | Fiber Optic dust cover                                              |      | (12-32004-01) |          |
| 8.   | 1   | F244 buffer 20 pin [surf mount]                                     | YES  | (19-20572-02) | .543     |
| 9.   | 1   | 10H116 ecl receiver [intrusive]                                     | YES  | (19-21936-01) | .432     |
| 10.  | 1   | 150 ohm resistor [surf mount]                                       | YES  | (13-23825-29) | nod      |
| 11.  | 6   | 60.4 ohm resistor [surf mount]                                      | YES  | (13-23827-76) | nod      |
| 12.  | 1   | 30 Pos Connector (with ejectors)                                    | YES  | (12-23982-02) |          |
| 13.  | 17  | .01 uF capacitor [surf mount]                                       | YES  | (10-24053-01) | .026     |
| 14.  | 2   | 3.3 uf capacitor [surf mount]                                       | YES  | (10-24455-13) | nod      |
| 15.  | 3   | 10 uf capacitor [surf mount]                                        | YES  | (10-24455-18) | .488     |
| 16.  | 1   | 1800 pf capacitor [surf mount]                                      | YES  | (10-24051-28) | nod      |
| 17.  | 1   | 3300 pf capacitor [surf mount]                                      | YES  | (10-24051-31) | nod      |
| 18.  | 9   | 196 ohm resistor [surf mount]                                       | YES  | (13-23828-29) | nod      |
| 19.  | 8   | 121 ohm resistor [surf mount]                                       | YES  | (13-23828-09) | nod      |
| 20.  | 1   | 383 ohm resistor [surf mount]                                       | YES  | (13-23828-57) | nod      |
| 21.  | 1   | 825 ohm resistor [surf mount]                                       | YES  | (13-23828-89) | nod      |
| 22.  | 2   | 1 uh inductor [intrusive]                                           | NO   | (16-32330-02) |          |
| 23.  | 1   | LED Green [intrusive]                                               | YES  | (11-14136-02) |          |

### C.3.2 FEATURES

- The Active Bulkhead Module has an on board LED, (green), to indicate a good connection to the main module and FDDI. The LED is visible from outside the cabinet.
- Controlled impedance etch on signal layers, (75 ohms nom).
- Controlled impedance cable along with heavy gauge power conductors allow error free operation with cable lengths of up to 15 feet.
- Module etch carrying high frequency signals, (125 Mhz), is sandwiched between power layers to provide good EMC .
- The 30 conductor cable is pinned such that plugging it in backwards will not cause damage to the adaptor.
- The 30 pin module connector comes with locking latches to prevent cable connection from becoming loose and causing a failure.
- Loopback mode permits adaptor, (main module and active bulkhead), testing without a fiber connection.
- Short circuit protection provided for with thermistors. The thermistors are located on the T2027 module.

### C.3.3 COMPONENT LOCATION

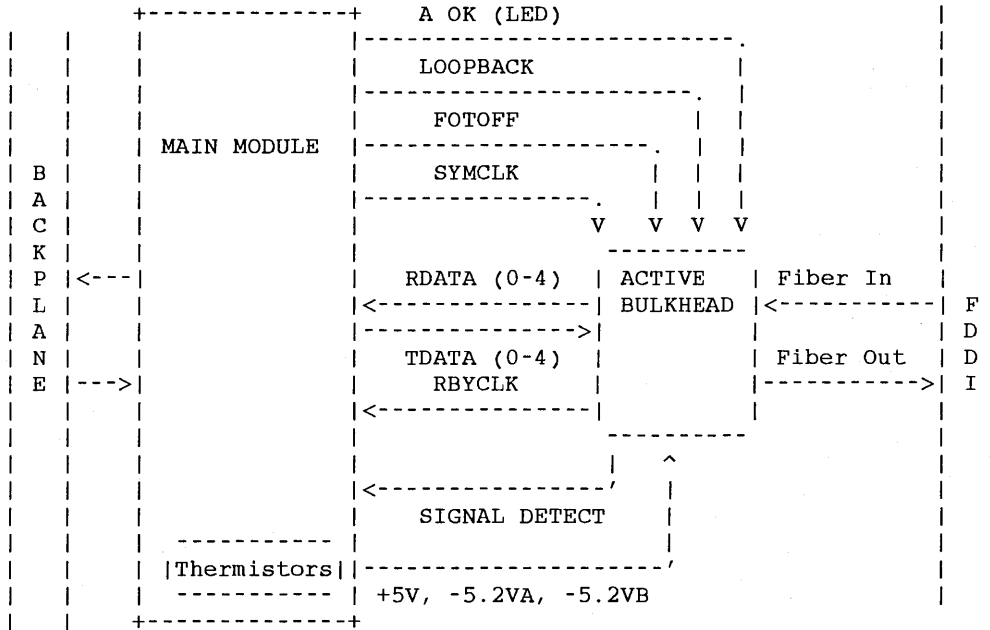
Figure 25: XMI to FDDI Adaptor Module Component Location



### C.3.4 DEMFA BLOCK DIAGRAM

The basic relationship of the XMI node, Active Bulkhead module, and FDDI is shown below.

Figure 26: Block Diagram of the XMI Node, Active Bulkhead, and FDDI.



### C.3.5 SIGNAL DEFINITIONS

INPUTS TO THE ACTIVE BULKHEAD MODULE (TOTAL=15)

TDATA0 thru TDATA4 - THESE ARE FIVE TTL LEVEL INPUTS. THIS IS THE 5 BIT PARALLEL DATA FROM THE MOTHER MODULE. THE DATA RATE IS 25 MHZ.

SYMCLK

FOTOFF

LOOPBACK

A OK (DRIVES STATUS LED)

-5.2VA (2 EACH)

-5.2VB (2 EACH)

+5.0V (2 EACH)

ACTIVE BULKHEAD MODULE OUTPUTS (TOTAL=7)

RDATA0 thru RDATA4 - THESE ARE FIVE TTL LEVEL OUTPUTS. THIS IS THE 5 BIT PARALLEL DATA TO THE MOTHER MODULE. THE DATA RATE IS 25 MHZ.

RBYCLK

SD (SIGNAL DETECT)

BIDIRECTIONAL SIGNALS

There are none

**C.3.6 TESTABILITY FEATURES**

The Active Bulkhead and its on-board components have a number of features that support testability.

- All surface-mounted resistors shall have test points accessible to bed-of-nails testers from side 2 of the module.
- All connector pins can be probed by testers from side 2 of the module.

**C.3.7 OPERATION**

For information on how the Active Bulkhead devices operate, refer to the Related Documents mentioned in Section 2 of this spec.

**C.4 RECOMMENDED OPERATING CONDITIONS**

**C.4.1 ENVIRONMENTAL CONDITIONS**

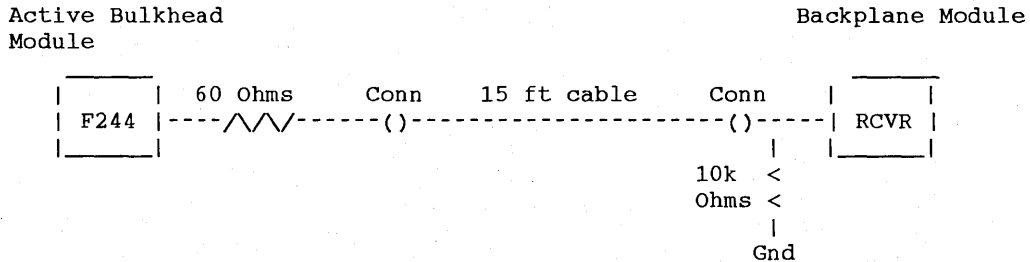
The Active Bulkhead module has an ambient storage temperature range of -40 degrees C to +80 degrees C (-40 degrees F to 176 degrees F). Storage relative humidity is 10% to 95%, with maximum wet bulb at 32 Degrees Centigrade and minimum dew point at 2 Degrees Centigrade.

The Active Bulkhead shall meet or exceed the requirements for operation within a system placed in a DEC Standard 102 Class B Environment.

**C.4.2 DC CHARACTERISTICS**

The +5.0 volt supply should not vary more than +/- 5%. The -5.2 volt supply should not vary more than +/- 5%. Maximum Module Power Dissipation 4.1 Watts

**C.4.3 TERMINATION REQUIREMENTS**



## C.5 DEVICE PINOUTS

### C.5.1 CONNECTOR PINOUT

- The following is a table containing the pin assignments of the 30 pin Active Bulkhead connector.

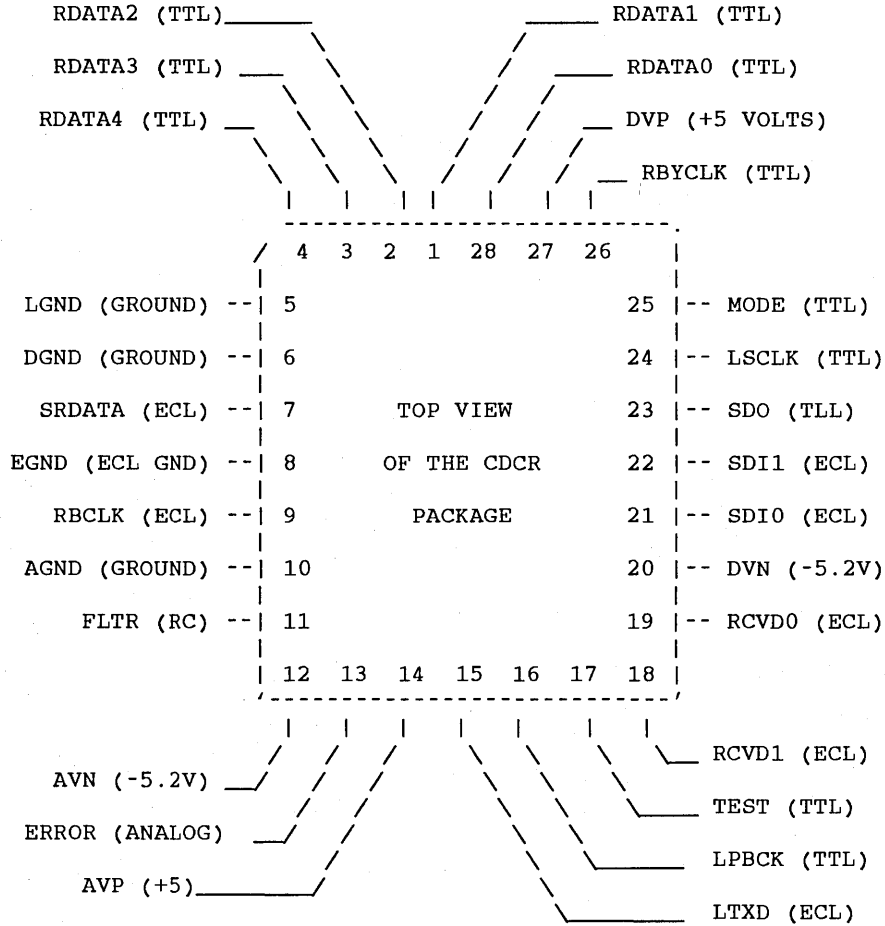
| PIN NUMBER | SIGNAL    | PIN NUMBER | SIGNAL    |
|------------|-----------|------------|-----------|
| 1          | GROUND    | 16         | GROUND    |
| 2          | +5.0V     | 17         | TDATA <0> |
| 3          | -5.2V     | 18         | RDATA <3> |
| 4          | C DETECT  | 19         | GROUND    |
| 5          | -5.2V     | 20         | RDATA <2> |
| 6          | RDATA <1> | 21         | TDATA <2> |
| 7          | GROUND    | 22         | GROUND    |
| 8          | RBYCLK    | 23         | TDATA <1> |
| 9          | FOTOFF    | 24         | LOOPBACK  |
| 10         | GROUND    | 25         | CSR       |
| 11         | SYMCLK    | 26         | -5.2V     |
| 12         | RDATA <0> | 27         | TDATA <4> |
| 13         | GROUND    | 28         | -5.2V     |
| 14         | RDATA <4> | 29         | +5.0V     |
| 15         | TDATA <3> | 30         | GROUND    |

- The table above shows the 8, (eight), ground connections required to limit the crosstalk to less than 100 millivolts.

**C.5.2 CDCR PINOUT**

The following diagram shows the connections to the CLOCK AND DATA CONVERSION RECEIVER, (CDCR).

**Figure 27: Block Diagram of the CDCR Connections**

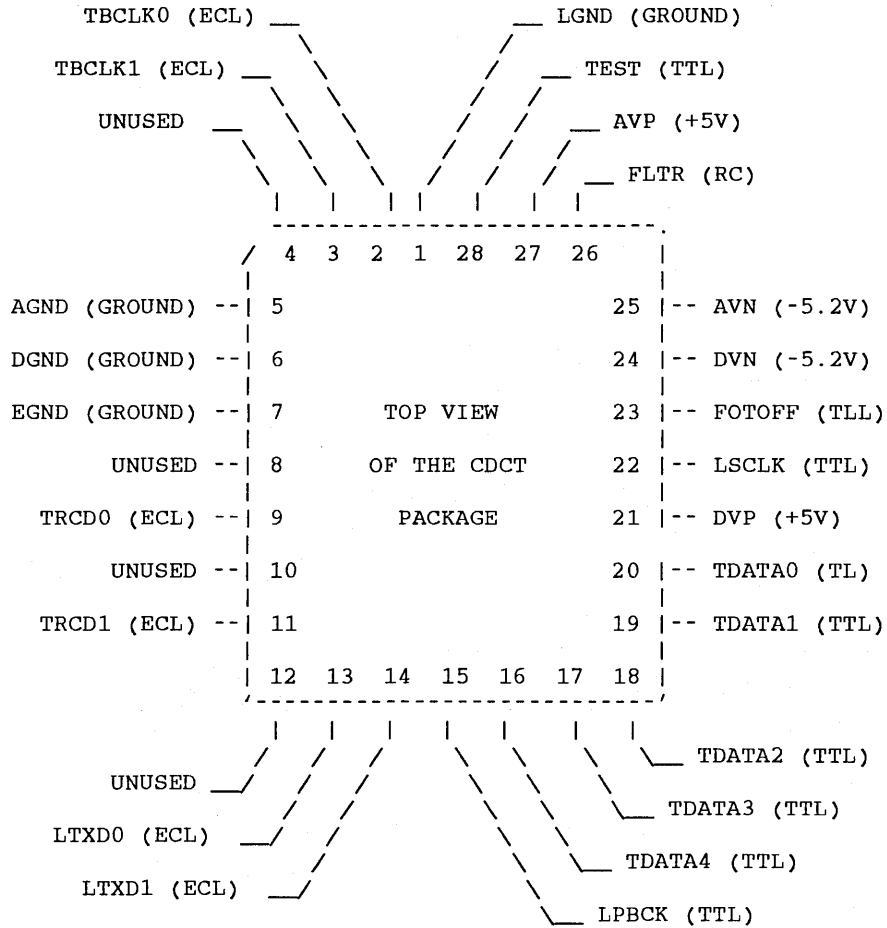




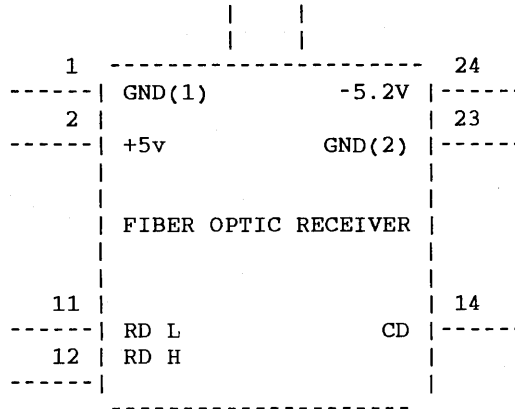
### C.5.3 CDCT PINOUT

The following diagram shows the connections to the CLOCK AND DATA CONVERSION TRANSMITTER, (CDCT).

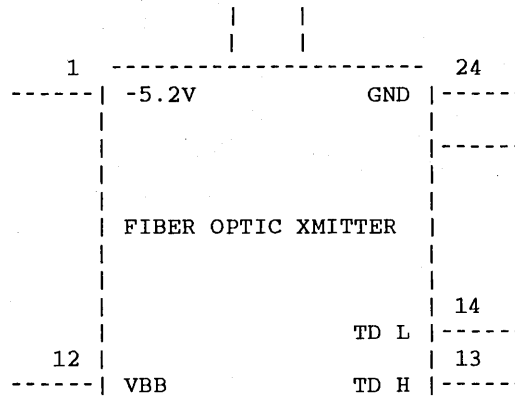
**Figure 28: Block Diagram of the CDCT Connections**



**C.5.4 FIBER OPTIC RCVR PINOUT**

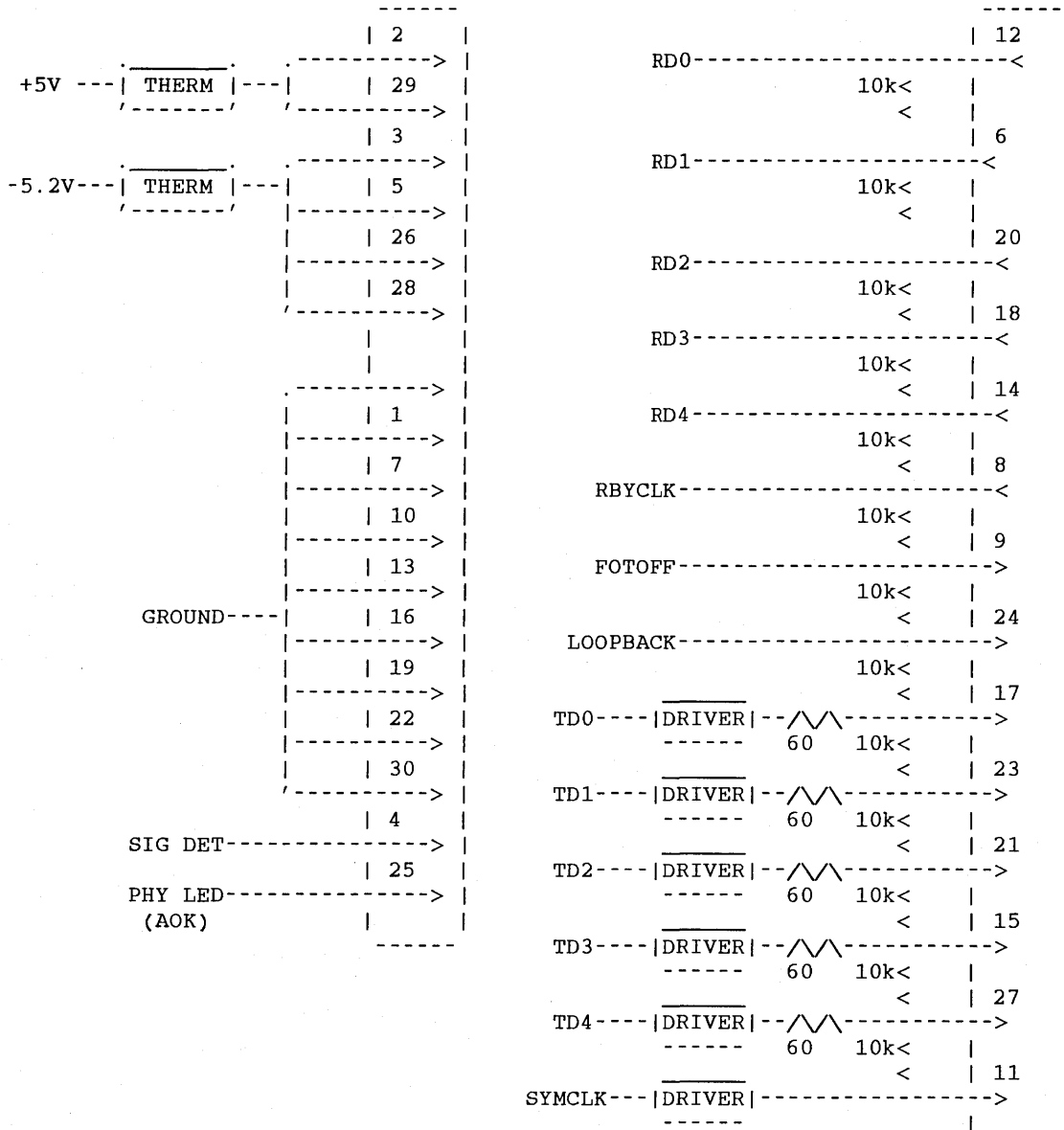


**C.5.5 FIBER OPTIC XMITTER PINOUT**



### C.6 T2027 MODULE REQUIREMENTS

- THERM, IS AN ABBREVIATION OF THERMISTOR, TO BE USED IN PLACE OF A FUSE.
- THE 10K RESISTORS, (13 EACH), ARE CONNECTED TO GROUND.



## C.7 ISSUES/MISCELLANEOUS

- Do we need a buffer to drive Rdata on cable. (Yes, the CDCR data outputs are not buffered)
- Do we need an ecl receiver for carrier detect. (Yes, for proper single ended to diferential drive conversion. We have good thermal compensation and voltage regulation using the 10H116.)
- Will we have to heat sink FOX chips and CDC chips. (we will use a conduction plate and reparable thermal epoxy for cdc chips only)
- We must have fuses or circuit breakers on the XMI module. ( Will be using two ther-mistors. One for Vcc and Vee).
- What termination will be used for cable. (60 ohm series, We are driving a single load at the end of a transmission line, signals are begging to be series terminated.
- Can outer layer 4 be solid power plane. (Yes, VLS will be happy to design this way.)
- Use SMOBC, Etch will not be plated.
- Should carrier detect be sent to the xmi module. (Yes, carrier detect will be sent to the XMI module.)
- Determine if LED to show status is necessary. Where should it reside. AB module or XMI module. (One green LED to be located on the Active Bulkhead module. Must be visible from back of cabinet.)
- Add pads and holes to place jumpers in place of inductor. (Will be using two 1uh intrusive inductors.)
- Holes size=(.040) will be added to the module so that rework wires can access both sides of the module. This is necessary because FOX and 10h116 pins can not be accessed on side 1.

**APPENDIX D**  
**PARSER ALGORITHM**

Figure 29: THE PARSING ALGORITHM - FC Filtering

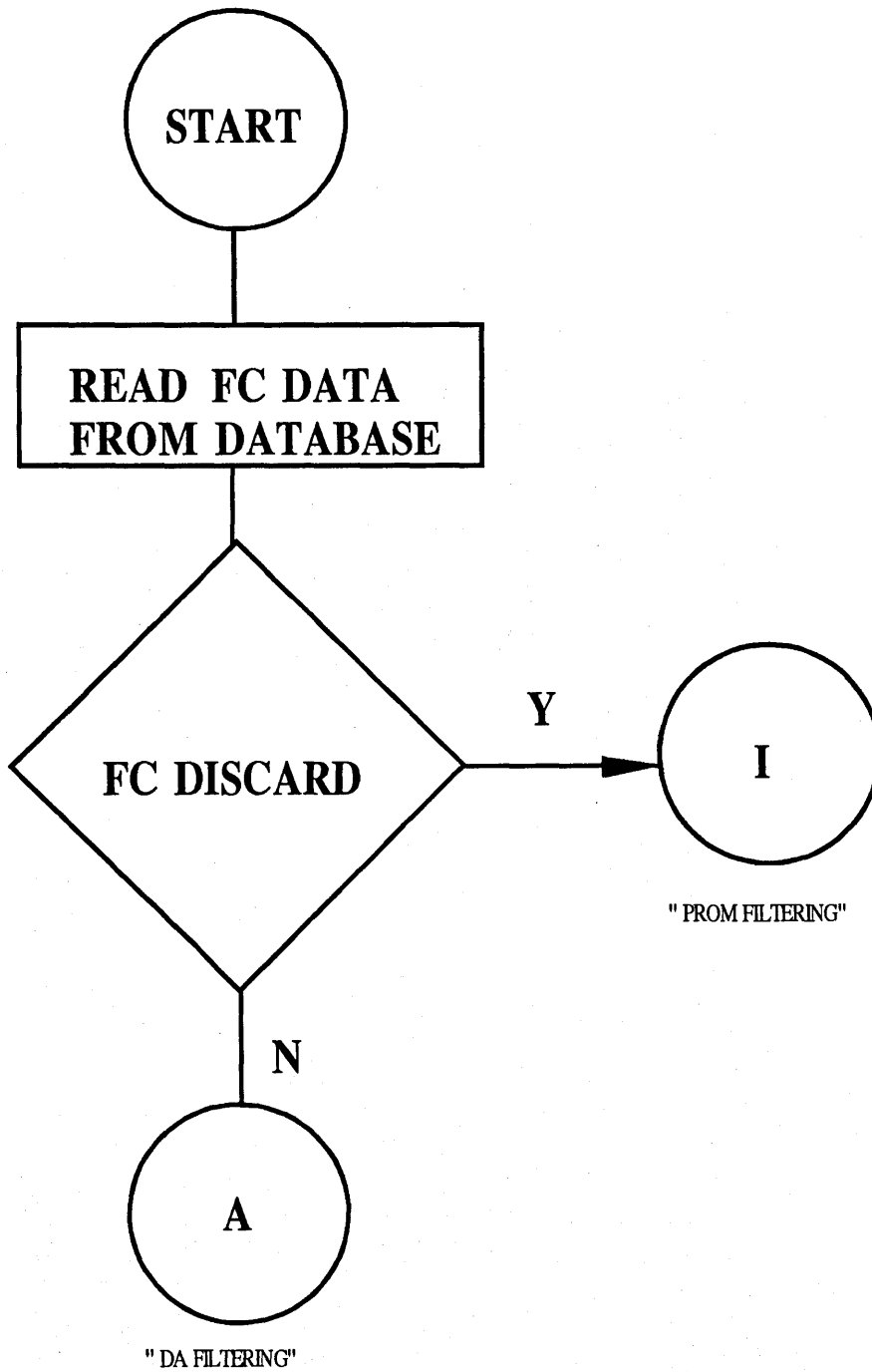


Figure 30: THE PARSING ALGORITHM (CONTD.) - DA Filtering

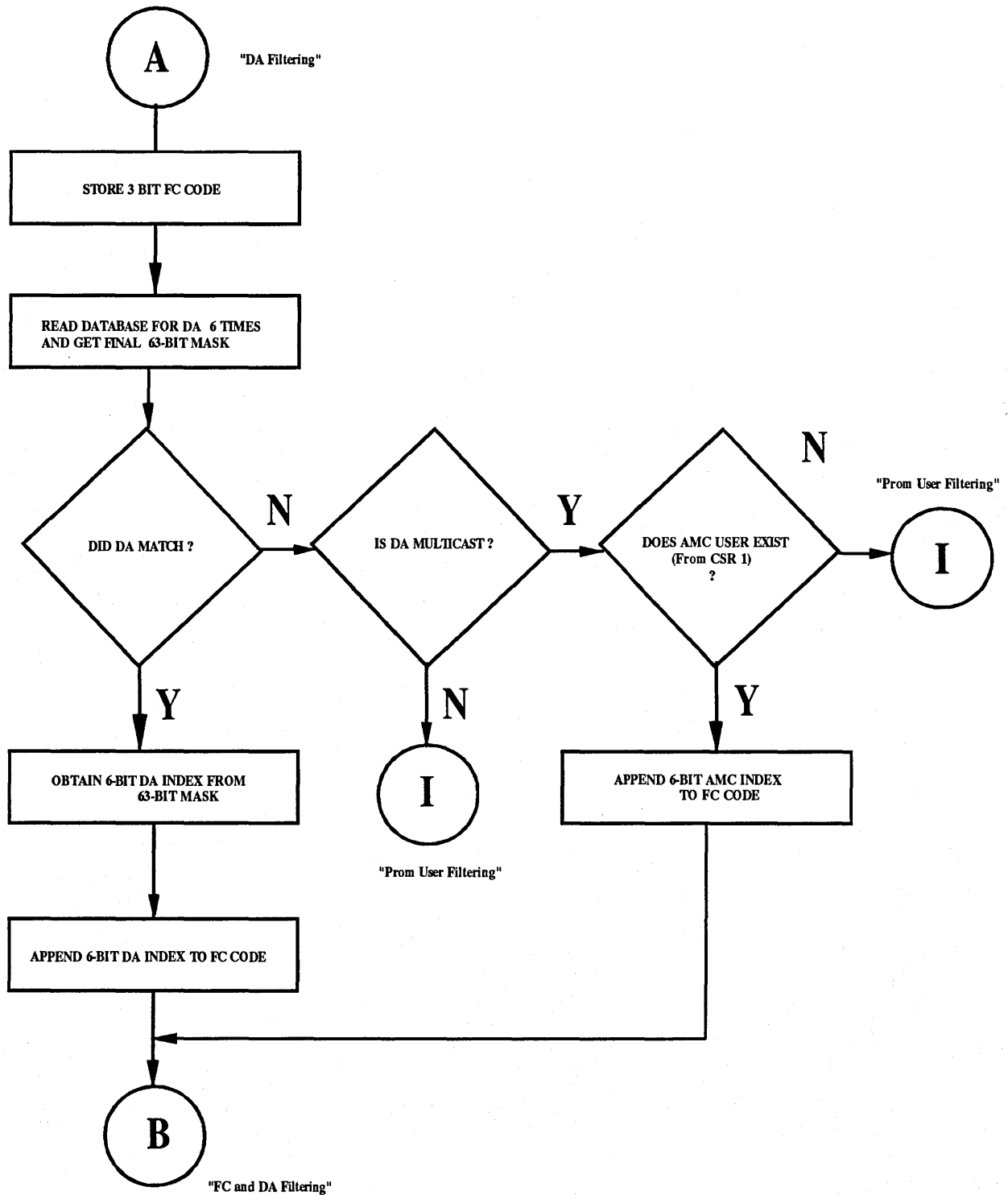


Figure 31: THE PARSING ALGORITHM (CONTD.) - FCDA Filtering

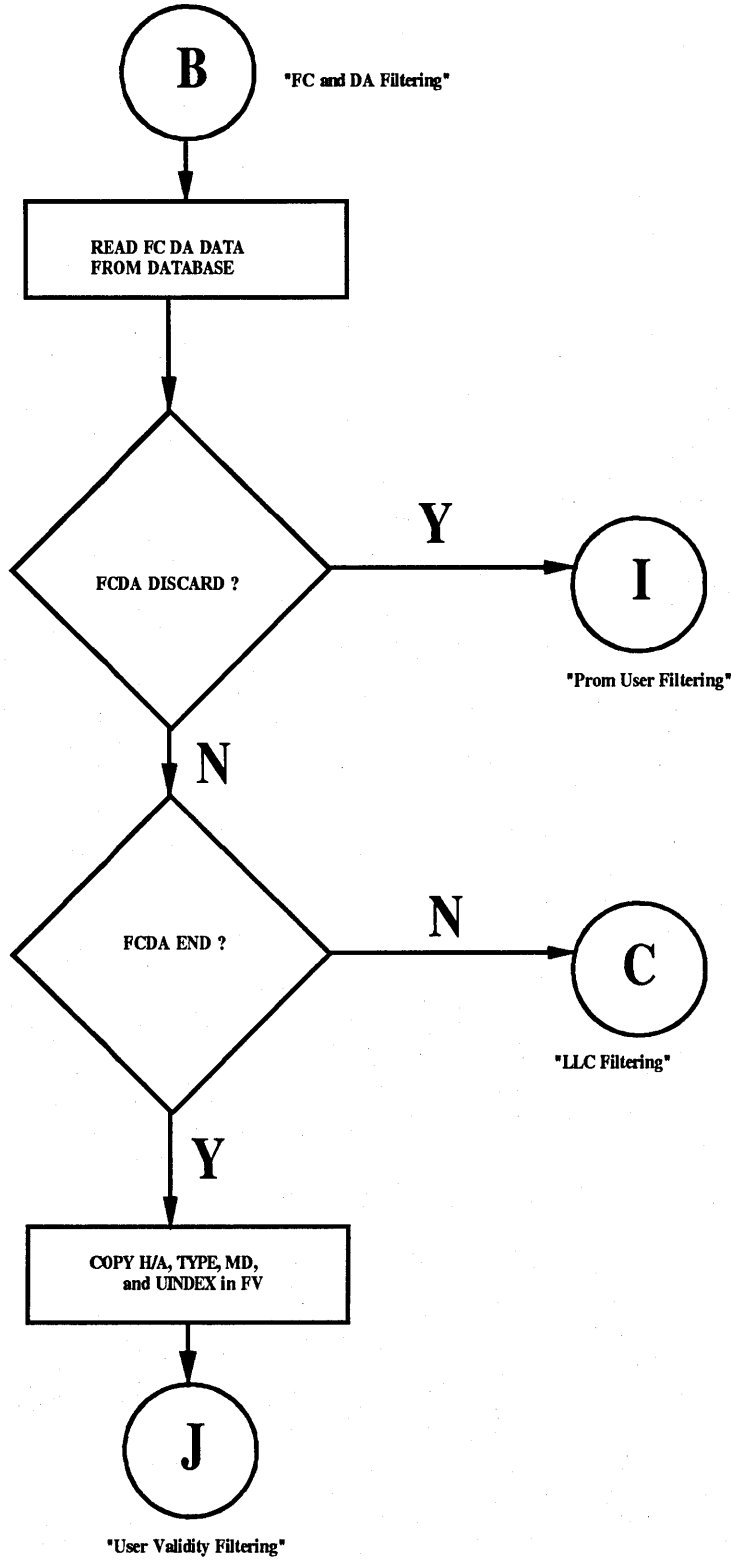
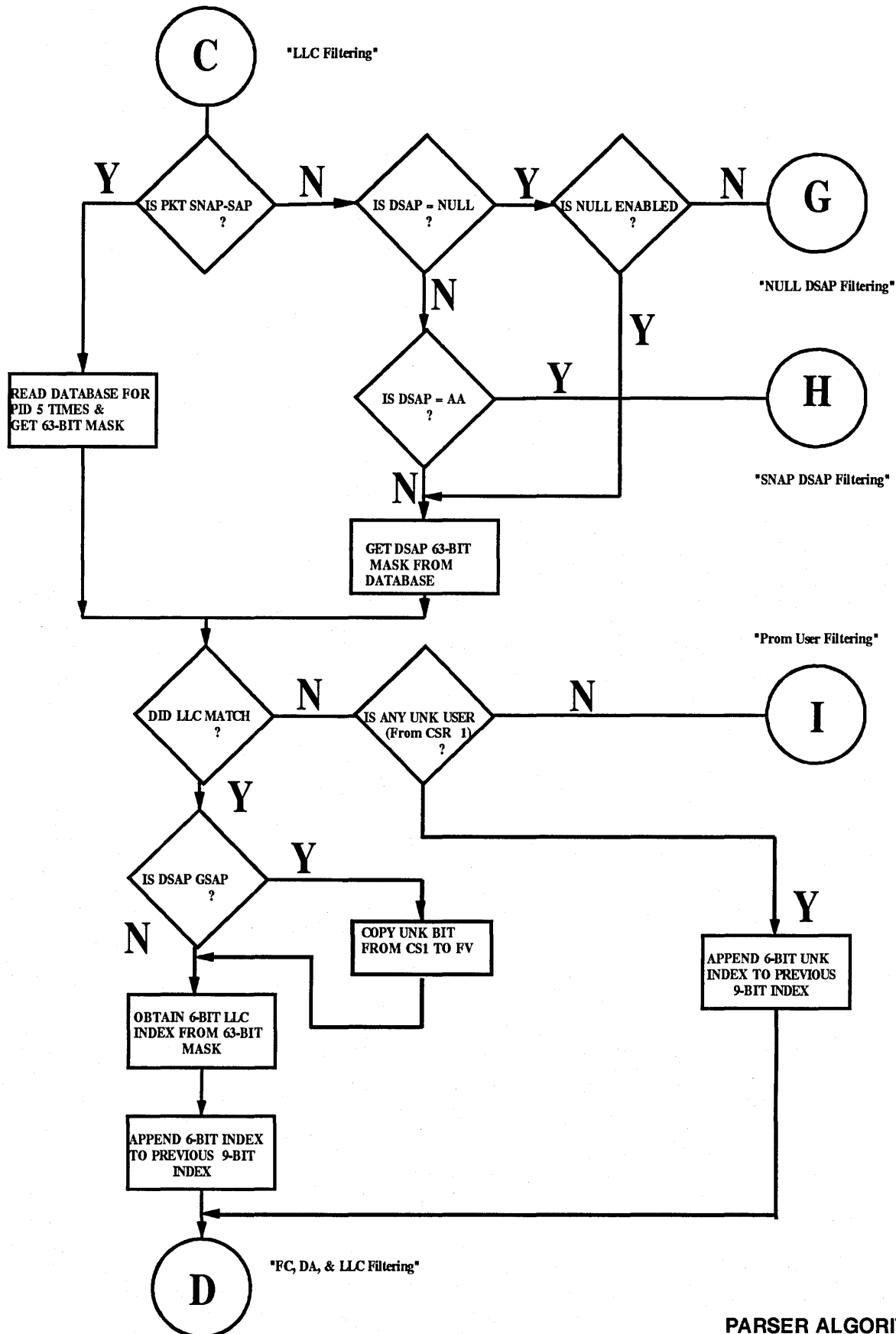




Figure 32: THE PARSING ALGORITHM (CONTD.) - LLC Filtering



Parsing Algorithm

Figure 33: THE PARSING ALGORITHM (CONTD.) - FCDALLC Filtering

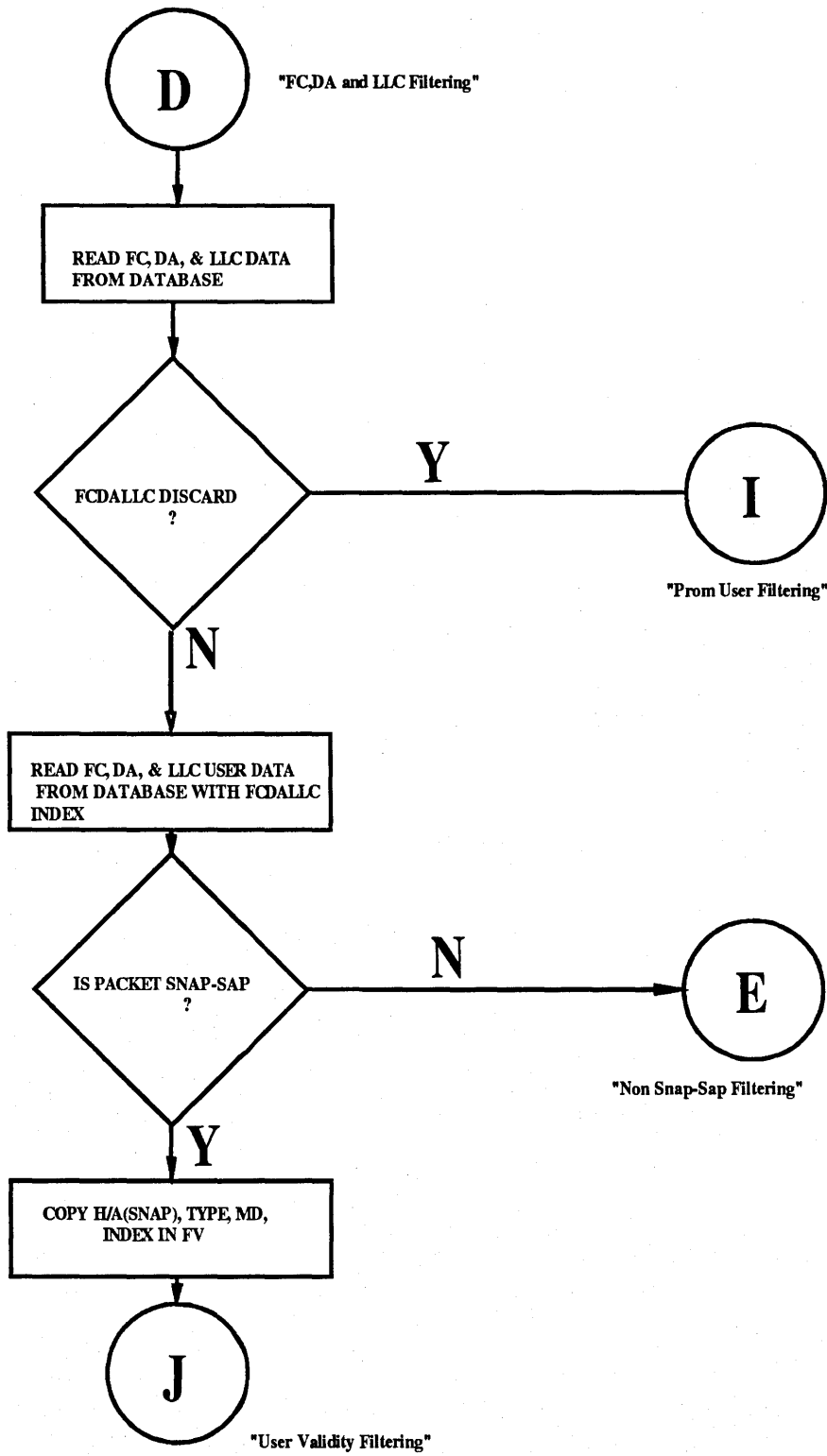


Figure 34: THE PARSING ALGORITHM (CONTD.) - Non SNAP SAP Filtering

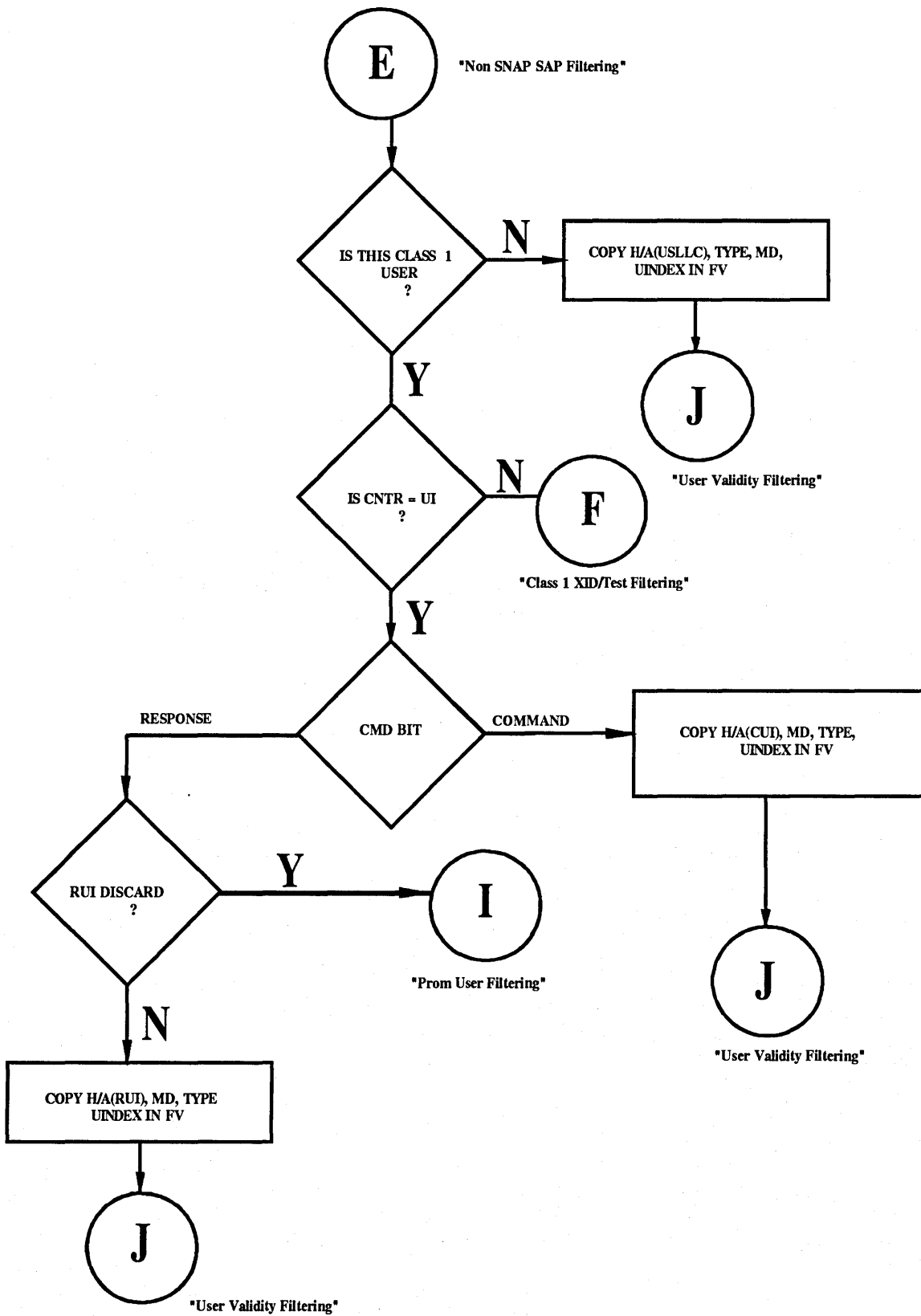


Figure 35: THE PARSING ALGORITHM (CONTD.) - Class1 XID/TEST Filtering

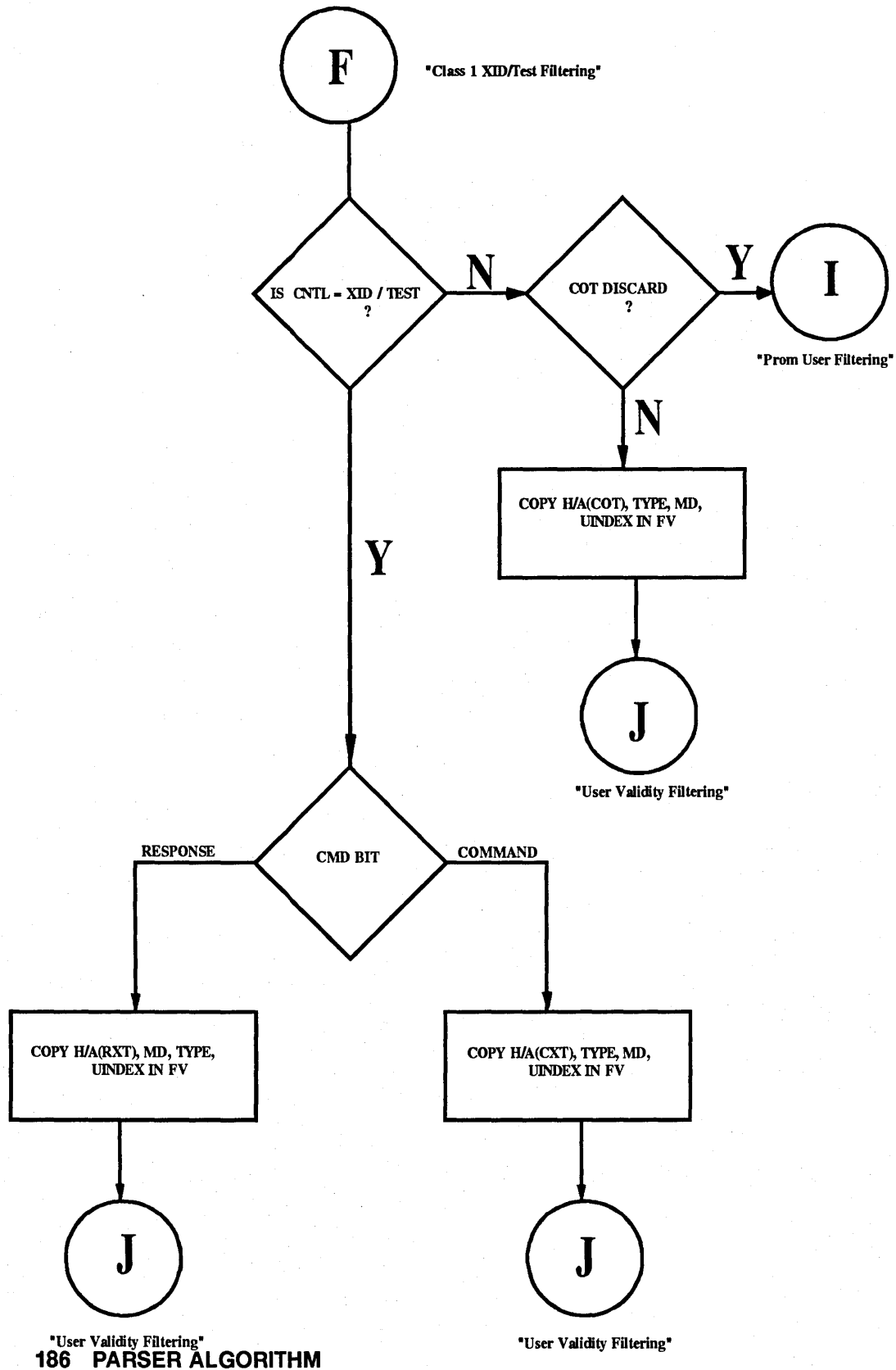


Figure 36: THE PARSING ALGORITHM (CONTD.) - NULL DSAP Filtering

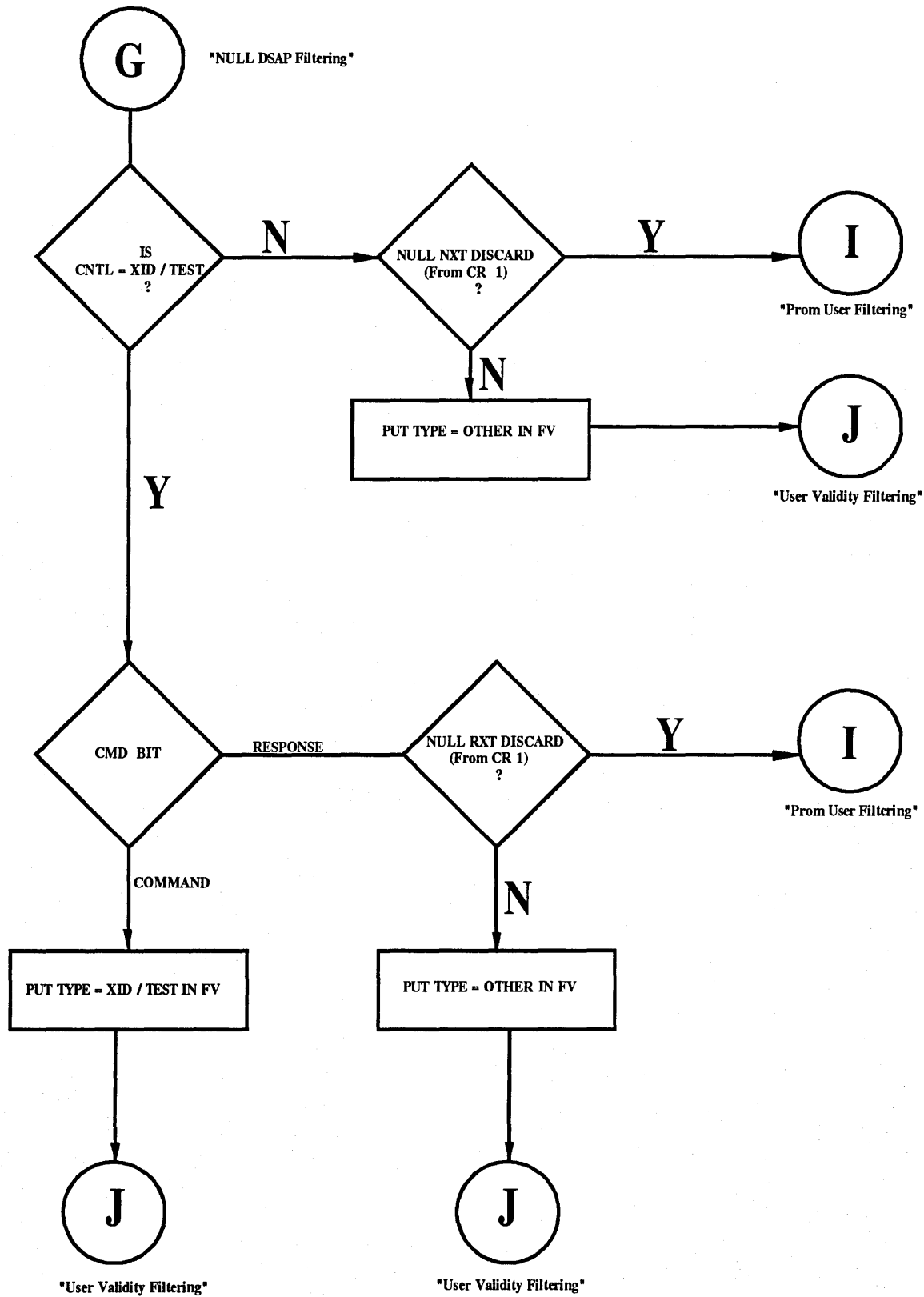


Figure 37: THE PARSING ALGORITHM (CONTD.) - SNAP DSAP Filtering

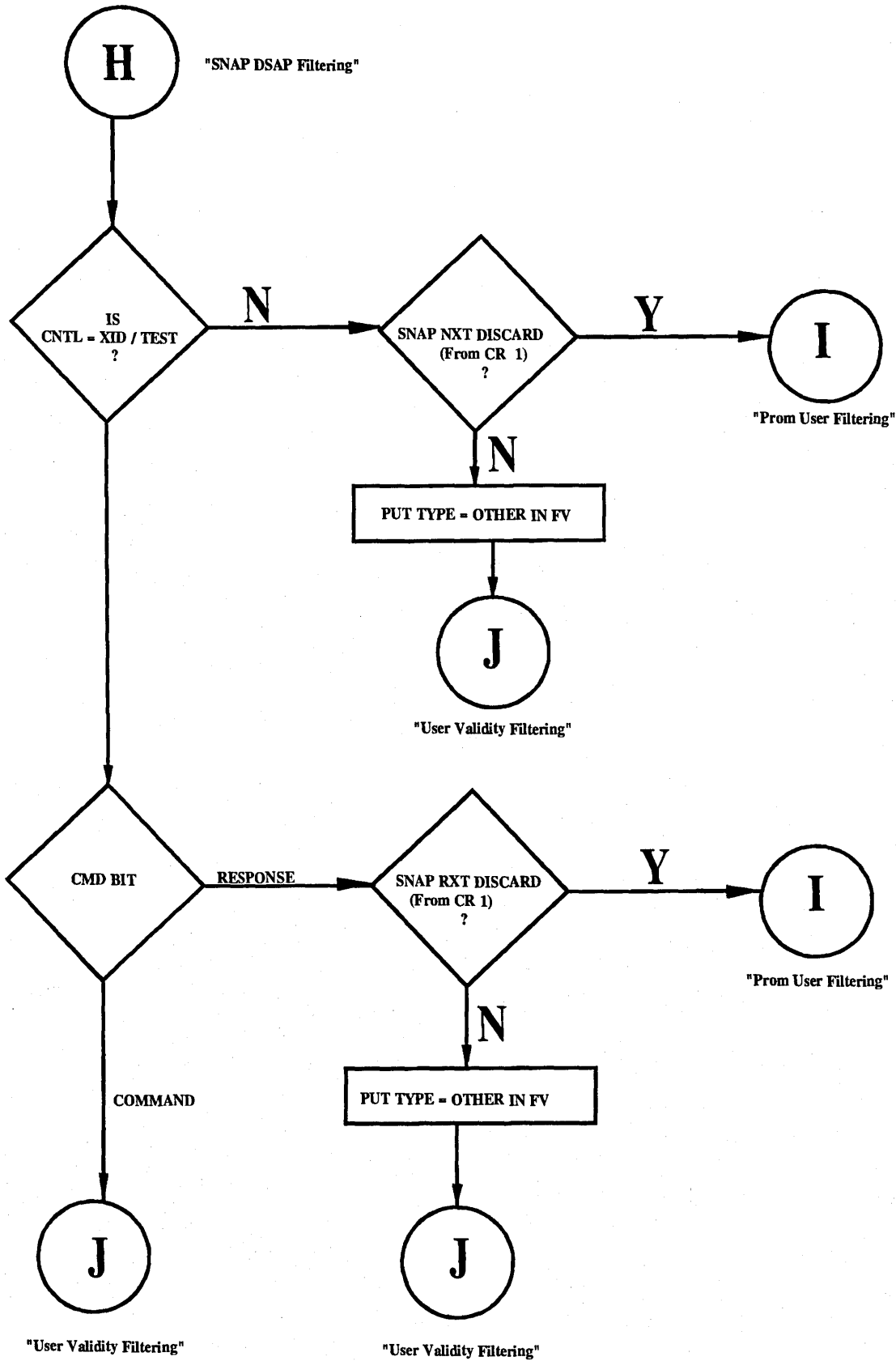


Figure 38: THE PARSING ALGORITHM (CONTD.) - Promiscuous User Filtering

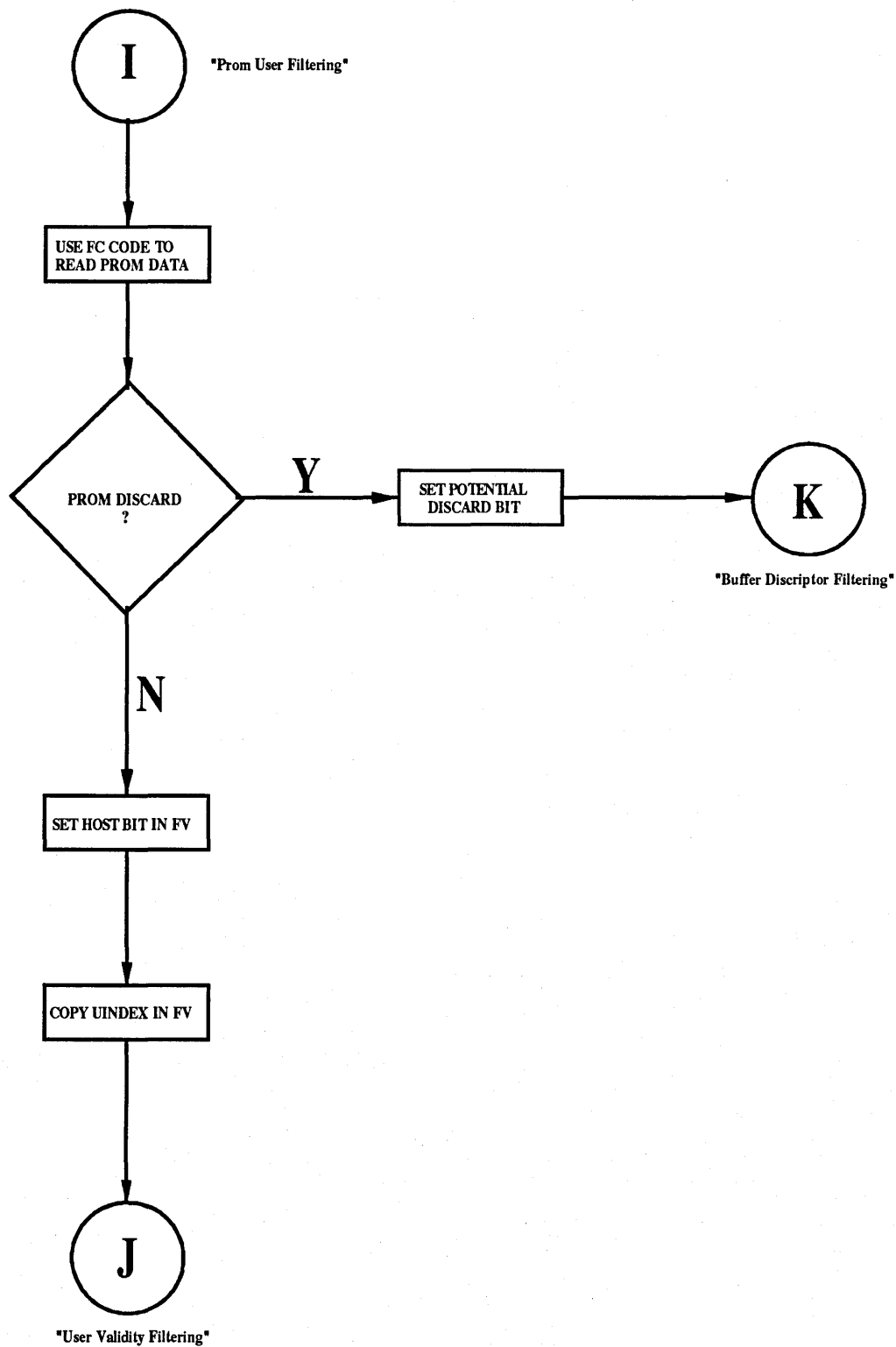


Figure 39: THE PARSING ALGORITHM (CONTD.) - User Validity Filtering

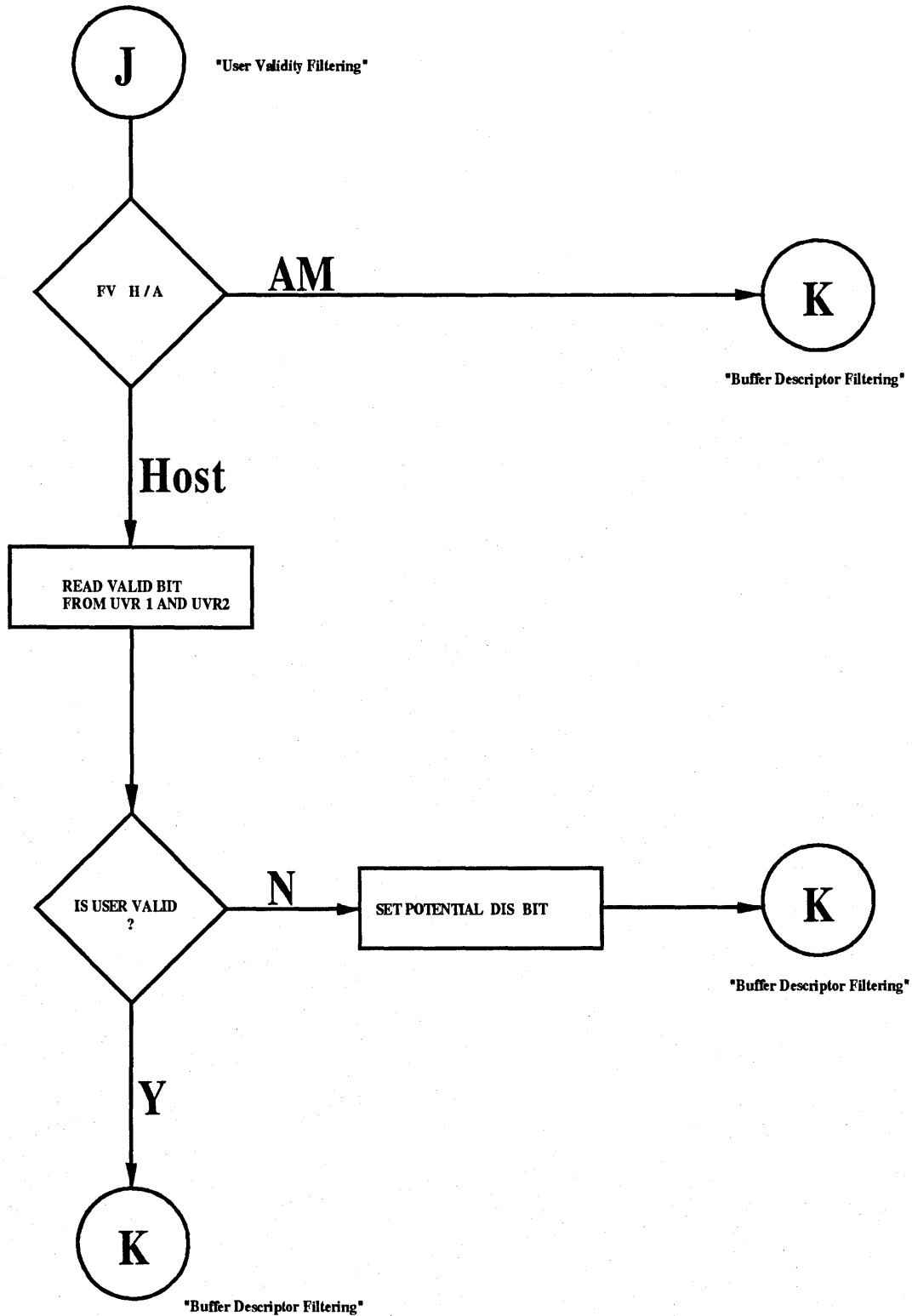
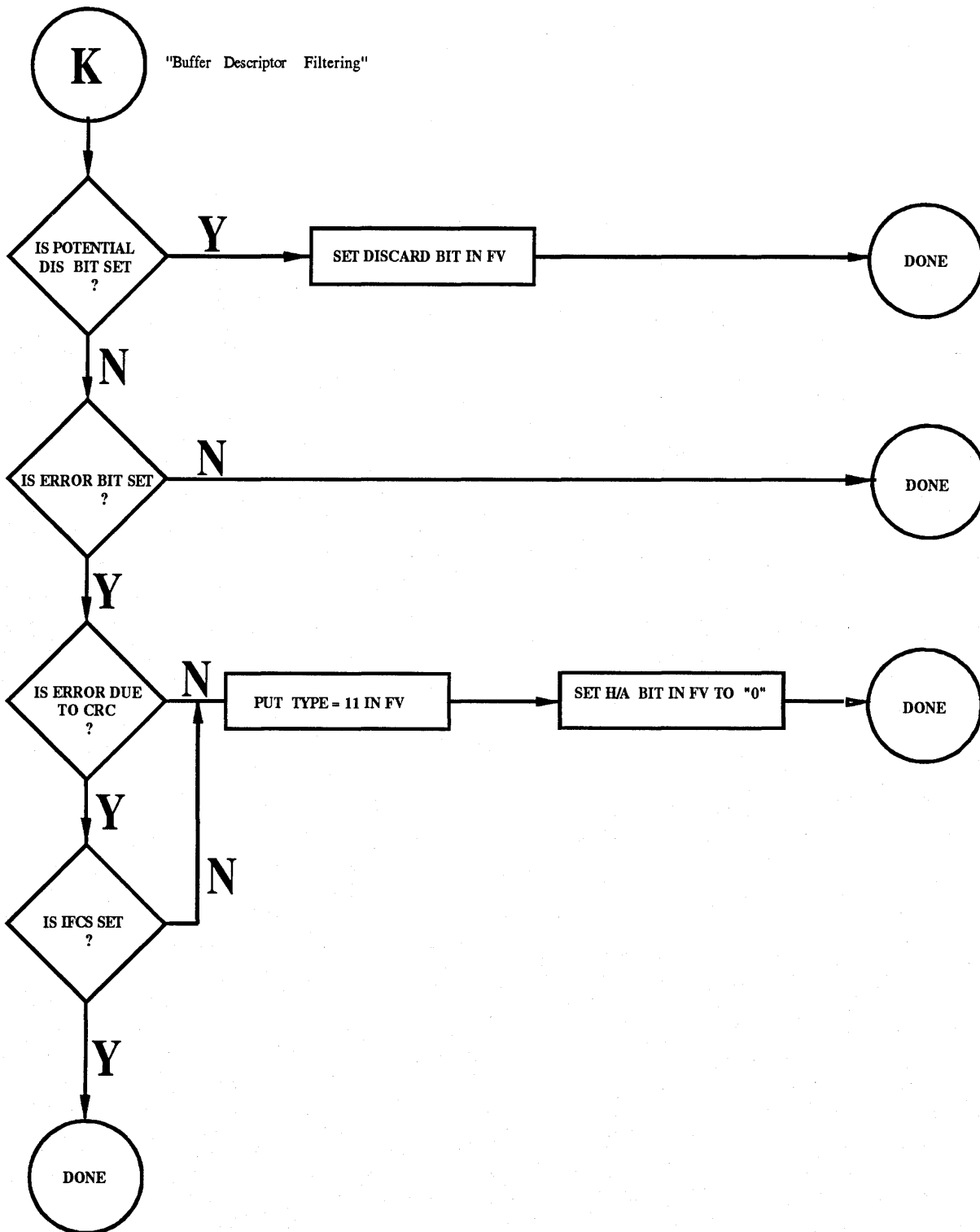




Figure 40: THE PARSING ALGORITHM (CONTD.) - Buffer Descriptor Filtering





COPY

Digital Equipment Corporation - Confidential and Proprietary

## AMI ( DC7200 ) Design Specification

\*\*\*\*\* PRELIMINARY \*\*\*\*\*

### Revision ??

This document specifies the design of the AMI Gate Array, as used on the XFA (XMI to FDDI) Adapter.

### Written by:

Dave Valley  
Dominic Gasbarro

VAX PRODUCTS AND OPTIONS

January 1989

Digital Equipment Corporation

# CONTENTS

|                                             |    |
|---------------------------------------------|----|
| <b>Chapter 1 EXECUTIVE SUMMARY</b> .....    | 3  |
| 1.1 OVERVIEW .....                          | 3  |
| 1.2 MICROPROCESSOR .....                    | 4  |
| 1.3 AMI GATE ARRAY .....                    | 5  |
| 1.3.1 Function Units .....                  | 6  |
| 1.4 68020 MEMORY SUBSYSTEM .....            | 8  |
| 1.5 SUBSYSTEM DATA INTEGRITY .....          | 8  |
| 1.6 POWER UP & SELF TEST .....              | 9  |
| <b>Chapter 2 AMI CONTEXT</b> .....          | 11 |
| 2.1 68020 INTERFACE .....                   | 12 |
| 2.2 LOCAL MEMORY INTERFACE .....            | 12 |
| 2.3 ESP INTERFACE .....                     | 13 |
| 2.4 PMC SIGNALS .....                       | 13 |
| 2.5 CSR BUS INTERFACE .....                 | 14 |
| 2.6 ATE TEST SIGNALS .....                  | 14 |
| 2.7 RESET INTERFACE .....                   | 15 |
| 2.8 XMI CORNER INTERFACE .....              | 15 |
| 2.9 AMI SYSTEM INTERFACE .....              | 15 |
| <b>Chapter 3 ADDRESS DECODE LOGIC</b> ..... | 17 |
| 3.1 OVERVIEW .....                          | 17 |
| 3.2 68020 INTERFACE .....                   | 19 |
| 3.3 LOCAL G.A. SIGNALS .....                | 19 |
| <b>Chapter 4 AMI CONTROL LOGIC</b> .....    | 23 |
| 4.1 OVERVIEW .....                          | 23 |
| 4.2 LOCAL INPUTS .....                      | 26 |
| 4.3 LOCAL OUTPUTS .....                     | 27 |
| 4.4 68020 CONNECTIONS .....                 | 27 |

|                  |                                                        |           |
|------------------|--------------------------------------------------------|-----------|
| <b>Chapter 5</b> | <b>RTOS, BUS ERROR &amp; WATCHDOG - TIMERS</b>         | <b>29</b> |
| 5.1              | OVERVIEW                                               | 29        |
| 5.2              | INPUT & OUTPUT SIGNALS                                 | 32        |
| 5.2.1            | WATCHDOG TIMER                                         | 32        |
| 5.2.1.1          | WATCHDOG COUNTERS                                      | 33        |
| 5.2.2            | BUS ERROR TIMER                                        | 33        |
| 5.2.2.1          | BUS ERROR COUNTER                                      | 33        |
| 5.2.3            | CONTROL CSR                                            | 33        |
| 5.2.4            | RTOS TIMER                                             | 33        |
| <br>             |                                                        |           |
| <b>Chapter 6</b> | <b>FDDI CHIPS CSR BUS INTERFACE</b>                    | <b>35</b> |
| 6.1              | OVERVIEW                                               | 35        |
| 6.1.1            | CSR BUS DATA PATH & SIGNALS                            | 37        |
| 6.1.2            | CSR INTERFACE GATE ARRAY INTERNAL SIGNALS              | 38        |
| 6.2              | CSR INTERFACE ADDRESSING                               | 39        |
| 6.3              | CSR INTERFACE BLOCK DESCRIPTION                        | 40        |
| 6.3.1            | SYNCHRONIZERS                                          | 41        |
| 6.3.2            | ADDRESS DECODER                                        | 41        |
| 6.3.3            | CSR BUS STATE MACHINE                                  | 41        |
| 6.3.4            | DIAGNOSTIC STATE MACHINE                               | 41        |
| 6.3.5            | DATA INPUT LATCH                                       | 41        |
| <br>             |                                                        |           |
| <b>Chapter 7</b> | <b>AMI/ESP INTERFACE</b>                               | <b>43</b> |
| 7.1              | OVERVIEW                                               | 43        |
| 7.1.1            | AMI/ESP INTERCONNECT BUS                               | 45        |
| 7.1.2            | AMI INTERNAL SIGNALS                                   | 45        |
| 7.1.3            | AMI/ESP INTERFACE FUNCTIONS                            | 46        |
| 7.1.4            | ADDRESS DECODER and CONTROL                            | 47        |
| 7.1.5            | PMC COMMAND/ADDRESS FORMATTER                          | 47        |
| 7.1.6            | PM READ - ADDRESS CACHE                                | 48        |
| 7.1.7            | AMI/ESP CYCLE CONTROL                                  | 48        |
| 7.1.8            | AMI/ESP STATUS REGISTER - ESPIFSR                      | 48        |
| 7.1.9            | AMI/ESP INTERFACE PARITY                               | 48        |
| 7.1.10           | LATCHES                                                | 50        |
| 7.1.11           | MUXES                                                  | 50        |
| 7.1.12           | PAGE OWNERSHIP                                         | 50        |
| 7.1.13           | CLEAR OWNERSHIP - DIRECT ADDRESSED PACKET MEMORY ONLY! | 50        |
| 7.1.14           | Buffer Descriptor, Page Table Entry and PMC CSR        | 51        |
| <br>             |                                                        |           |
| <b>Chapter 8</b> | <b>INTERRUPT CONTROLLER</b>                            | <b>53</b> |
| 8.1              | OVERVIEW                                               | 53        |
| 8.1.1            | Interrupt Sources                                      | 53        |
| 8.1.2            | INTERRUPT SOURCE SIGNALS                               | 55        |

|                                                |           |
|------------------------------------------------|-----------|
| 8.1.3 INTERRUPT CONTROLLING SIGNALS .....      | 55        |
| 8.1.4 Priority Encoder Logic .....             | 58        |
| 8.1.5 Address Decode and Control .....         | 58        |
| 8.1.6 Interrupt Acknowledge Logic .....        | 59        |
| 8.1.7 Self-Test Register .....                 | 59        |
| 8.1.8 CSR Registers .....                      | 59        |
| 8.1.9 Interrupt Mask Register .....            | 59        |
| 8.1.10 Interrupt Enable/Disable Register ..... | 59        |
| 8.1.11 Vector Registers .....                  | 60        |
| 8.2 Nature and Handling of Interrupts .....    | 60        |
| 8.3 Interrupt Latency .....                    | 60        |
| <b>Chapter 9 RESET LOGIC .....</b>             | <b>63</b> |
| 9.1 OVERVIEW .....                             | 63        |
| <b>Chapter 10 MEMORY SUBSYSTEM .....</b>       | <b>65</b> |
| 10.1 BUFFERS & TRANSCEIVERS .....              | 65        |
| 10.2 MEMORY SUBSYSTEM CONTROL .....            | 65        |
| 10.3 SRAM .....                                | 65        |
| 10.4 EEPROM .....                              | 65        |
| 10.5 DPA PROM .....                            | 65        |
| 10.6 SRAM PARITY .....                         | 65        |
| 10.7 SRAM & EEPROM READDRESSING .....          | 66        |
| <b>Chapter 11 PARITY LOGIC .....</b>           | <b>69</b> |
| 11.1 OVERVIEW .....                            | 69        |
| <b>Chapter 12 SELF-TEST LOGIC .....</b>        | <b>73</b> |
| 12.1 OVERVIEW .....                            | 73        |
| 12.1.1 Memory Testing .....                    | 73        |
| 12.1.1.1 Local SRAM .....                      | 73        |
| 12.1.1.2 Local EEPROM .....                    | 73        |
| 12.1.1.3 Local DPA .....                       | 73        |
| 12.1.1.4 Packet Memory Testing .....           | 74        |
| 12.1.2 Parity Logic Testing .....              | 74        |
| 12.1.3 Hardware & Watchdog Timers .....        | 74        |
| 12.1.4 AMI/ESP Interface .....                 | 74        |
| 12.1.5 CSR Bus Interface .....                 | 74        |
| 12.1.6 Interrupt Logic .....                   | 74        |
| 12.1.7 DSACK & BUS ERROR Logic .....           | 74        |
| 12.1.8 Address Decode .....                    | 75        |

|                                                                         |            |
|-------------------------------------------------------------------------|------------|
| 12.1.9 Reset Logic .....                                                | 75         |
| 12.1.10 Memory Subsystem Control Logic .....                            | 75         |
| <b>Chapter 13 LOGIC SUPPORTING ERROR CONDITIONS .....</b>               | <b>77</b>  |
| 13.1 OVERVIEW .....                                                     | 77         |
| 13.1.1 AMI/ESP ERROR STATUS LINES .....                                 | 77         |
| 13.1.2 BUS ERROR CONDITIONS .....                                       | 77         |
| 13.1.2.1 IMPROPER ADDRESS ALIGNMENT .....                               | 77         |
| <b>Chapter 14 CONTROL STATUS REGISTERS &amp; MISC. GLUE LOGIC .....</b> | <b>79</b>  |
| 14.1 OVERVIEW .....                                                     | 79         |
| 14.1.1 CLOCKS INTERFACE .....                                           | 79         |
| 14.1.2 AMI CSRs .....                                                   | 79         |
| 14.1.3 SPECIAL CSR ACCESSES .....                                       | 79         |
| <b>Chapter 15 UPDATE (02-19-1990) AMI REGISTERS .....</b>               | <b>81</b>  |
| 15.1 ESP INTERFACE STATUS REGISTER (ESPIFSR) .....                      | 81         |
| 15.2 TIMERS AND COUNTERS .....                                          | 82         |
| 15.2.1 AMI WATCHDOG TIMER (WDTIMR) .....                                | 82         |
| 15.2.2 AMI RTOS TIMER (RTOSTIMR) .....                                  | 83         |
| 15.2.3 COUNTER/TIMER CSR (CTCSR) .....                                  | 84         |
| 15.2.4 BUS ERROR TIMER (BET) .....                                      | 85         |
| 15.3 AMI GENERAL PURPOSE CSR (GPCSR) .....                              | 86         |
| 15.4 AMI EVENT STROBES .....                                            | 89         |
| 15.4.1 ADAPTER MANAGER TRANSMIT DONE STROBE (AMTDS) .....               | 89         |
| 15.4.2 ADAPTER MANAGER RECEIVE DONE STROBE (AMRDS) .....                | 90         |
| 15.4.3 SMT PRIORITY STROBE (SMTPS) .....                                | 90         |
| 15.4.4 RTOS EVENT STROBE (RTOSES) .....                                 | 91         |
| 15.4.5 WATCHDOG TIMER CLEAR STROBE (WDTCLR) .....                       | 91         |
| 15.5 AMI MISCELANEOUS REGISTERS .....                                   | 92         |
| 15.5.1 PARITY ERROR ADDRESS REGISTER (PEAR) .....                       | 92         |
| 15.6 INTERRUPT AND INTERRUPT VECTOR REGISTERS .....                     | 92         |
| 15.6.1 INTERRUPT CSR 1 (INTCSR1) .....                                  | 92         |
| 15.6.2 INTERRUPT CSR 2 (INTCSR2) .....                                  | 94         |
| 15.6.3 SOURCE INTERRUPT MASK REGISTER (SIMR) .....                      | 96         |
| 15.6.4 IPL INTERRUPT MASK REGISTER (IPLIMR) .....                       | 97         |
| 15.6.5 SOURCE INTERRUPT ENABLE REGISTER (SIER) .....                    | 99         |
| 15.6.6 IPL INTERRUPT ENABLE REGISTER (IPLIER) .....                     | 100        |
| 15.6.7 INTERRUPT SELF TEST REGISTER (INTST) .....                       | 101        |
| 15.6.8 VECTOR REGISTERS (VECREGS) .....                                 | 104        |
| <b>Appendix A ESP GATE ARRAY REGISTERS VISIBLE TO THE AMI .....</b>     | <b>111</b> |

## FIGURES

|    |                                            |    |
|----|--------------------------------------------|----|
| 1  | 68020 SUBSYSTEM - BLOCK DIAGRAM .....      | 4  |
| 2  | AMI BLOCK DIAGRAM .....                    | 5  |
| 3  | AMI CONTEXT .....                          | 11 |
| 4  | ADDRESS DECODER - BLOCK DIAGRAM .....      | 17 |
| 5  | AMI_ADR - 2ND LEVEL BLOCK DIAGRAM .....    | 18 |
| 6  | ADDRESS MAP - BLOCK DIAGRAM .....          | 21 |
| 7  | AMI CONTROL LOGIC - BLOCK DIAGRAM .....    | 24 |
| 8  | AMI_CTRL - 2ND LEVEL BLOCK DIAGRAM .....   | 25 |
| 9  | COUNTER/TIMER .....                        | 29 |
| 10 | AMI_TMRS - 2ND LEVEL BLOCK DIAGRAM .....   | 30 |
| 11 | EXAMPLE OF EACH COUNTER STAGE .....        | 31 |
| 12 | CSR INTERFACE CONTEXT .....                | 36 |
| 13 | CSR INTERFACE - BLOCK DIAGRAM .....        | 40 |
| 14 | AMI/ESP INTERFACE CONTEXT .....            | 44 |
| 15 | AMI/ESP Transaction Registers .....        |    |
|    | Registers contained within ESP Chip .....  | 47 |
| 16 | ESP INTERFACE - BLOCK DIAGRAM .....        | 49 |
| 17 | INTERRUPT CONTROL CONTEXT .....            | 54 |
| 18 | INTERRUPT CONTROL DETAIL BLOCK .....       | 57 |
| 19 | Reset Logic Block Diagram .....            | 64 |
| 20 | 68020 Memory Subsystem Block Diagram ..... | 67 |
| 21 | PARITY CIRCUIT DIAGRAM .....               | 70 |
| 22 | PARITY CIRCUIT DIAGRAM .....               | 71 |

## TABLES

|   |                                     |    |
|---|-------------------------------------|----|
| 1 | INTERRUPT STRUCTURE .....           | 7  |
| 2 | CSR TO BE ACCESSED .....            | 35 |
| 3 | CSR BUS - FDDI CHIP SELECTION ..... | 39 |
| 4 | INTERRUPT STRUCTURE .....           | 58 |
| 5 | Worst Case Interrupt Latency .....  | 60 |
| 6 | Typical Interrupt Latency .....     | 61 |





# CHAPTER 1

## EXECUTIVE SUMMARY

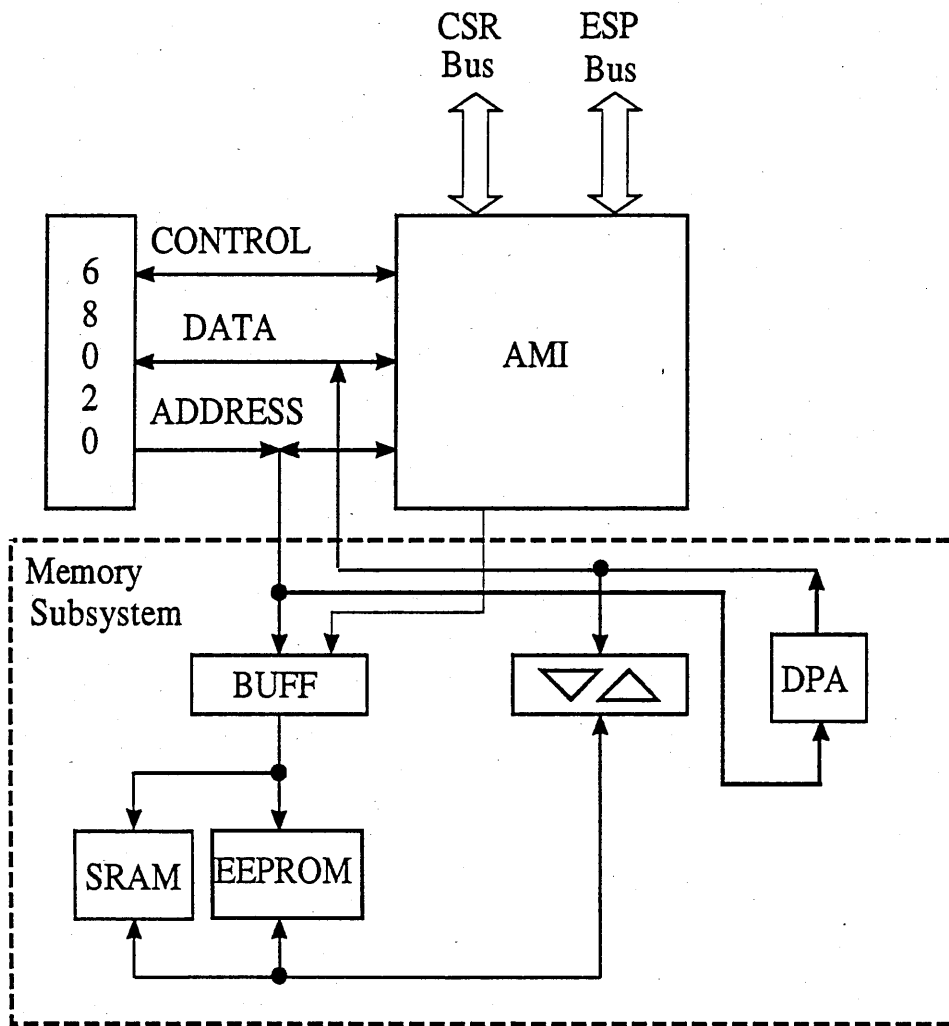
### 1.1 OVERVIEW

The Adapter Manager will act as a Node Processor executing code to perform the following tasks:

- Self Test
- SMT Frame Processing ( SMT/CMT )
  - FDDI Physical Initialization & Control
  - FDDI Logical Initialization & Control
- XID and TEST Processing
- Remote Console Frame Processing ( MOP )
- Loopback ( MOP )
- Adapter Control via Port Driver Commands

The Adapter Manager will be a processing entity based on the Motorola MC68020 32-bit Microprocessor.

Figure 1: 68020 SUBSYSTEM - BLOCK DIAGRAM



## 1.2 MICROPROCESSOR

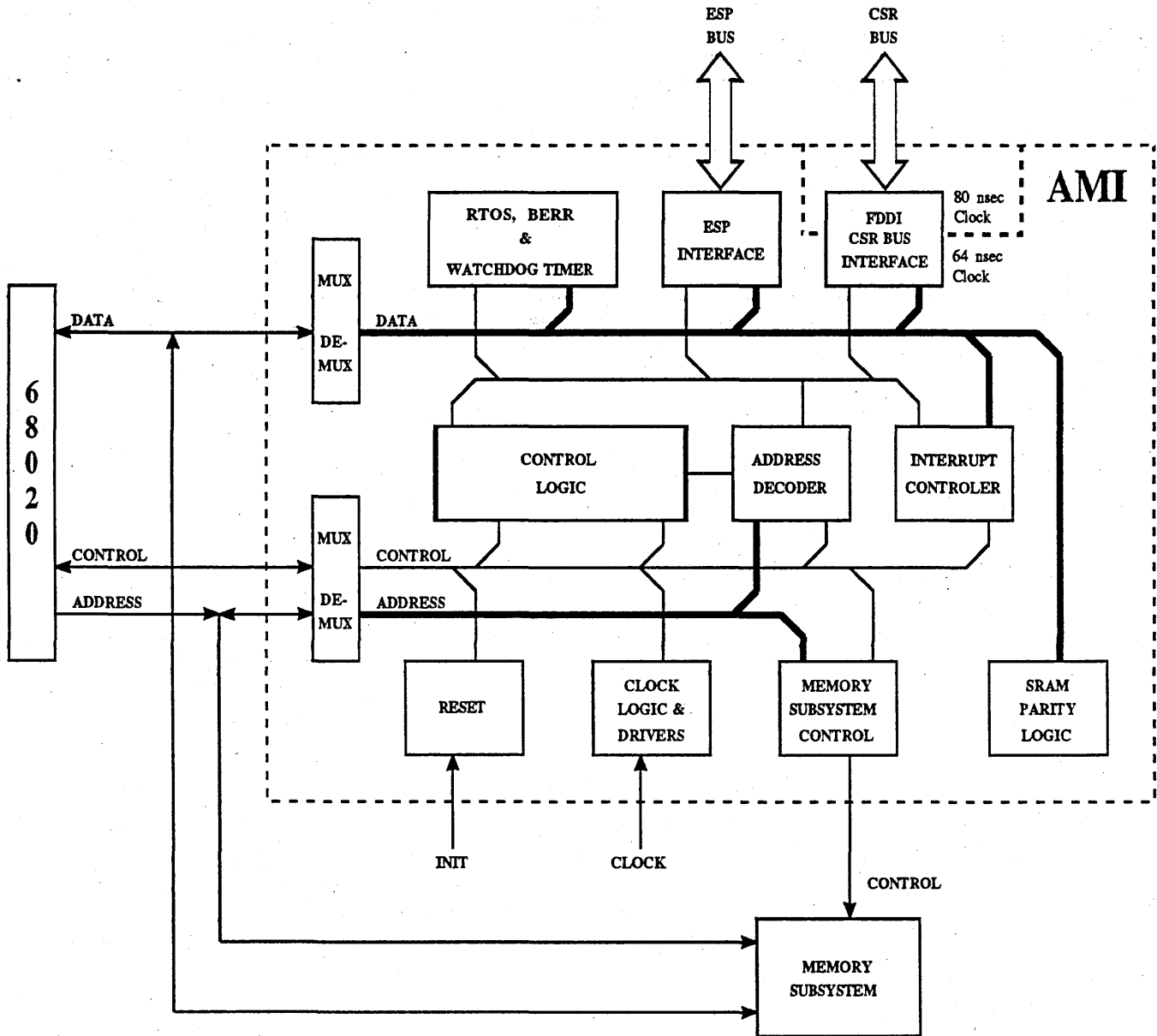
The microprocessor chosen for the Adapter Manager is a 16.67Mhz Motorola MC68020. The MC68020 will be run on a 50% Duty Cycle 15.625Mhz ( 64nsec ) clock derived from the XMI Corner XCI 34H Clock and circuitry within the AMI. Single stepping the microprocessor will not be supported in hardware, but can be achieved with an In Circuit Emulator. The MC68020 is a high performance microprocessor implemented in HCMOS technology and contains 32-bit registers and data paths, 32-bit physical addressing, on chip cache, dynamic

bus sizing and a pipelined architecture. At 15.625Mhz the 68020 offers approximately 2 MIPs performance. The device used in this implementation is packaged in a 114 Pin PGA.

### 1.3 AMI GATE ARRAY

The AMI Gate Array will combine all the necessary function units and logic, with the exception of memory, to support the 68020 microprocessor.

Figure 2: AMI BLOCK DIAGRAM



### 1.3.1 Function Units

- **Address Decoder:** Physical addresses produced by the 68020 are decoded in this unit to generate the enable signals for memory and peripheral devices within the Gate Array. This function unit will also support the control signals for dynamic bus sizing.
- **CONTROL Logic:** This logic in co-ordination with the Decode Logic will provide the 68020 with the DSACK signals required to notify the 68020 that a data transfer operation is complete. DSACK signals must be generated during interrupt acknowledge cycles as well. This Data Acknowledge function block will also contain a wait state generator to add machine cycles for slow peripheral accesses such as to EEPROM.
- **RTOS, BERR and WATCHDOG Timer:** This function unit will contain the RTOS Timer that provides the periodic interrupt for the RTOS Real Time Clock. Estimates on the periodicity of the timer output to be programmed are in the range of 5msec to 10msec. The RTOS timer output will be programmable to a maximum period of 1 second.

A Bus Cycle Timer will be included in this section to provide a Bus Error signal to the 68020 if a bus cycle does not complete in a specified period of time. Upon receipt of a Bus Error the 68020 will trap to a error handling routine. This is to prevent a "hung" 68020 bus in the event of an error. The Bus Error Counter can be programmed for up to 4 milliseconds before expiring if no DSACKL is received.

The Watchdog Timer will provide a keep alive function for the 68020. The Watchdog Timer is periodically cleared by a 68020 CSR access under software control. In the event that software branches off in an uncontrolled manner ( failure ), the Watchdog Timer will time out and generate an interrupt. The interrupt will cause the 68020 to vector to an Error Handling Routine. In addition, dedicated error reporting signals XERCOD<1:0> between the AMI and the ESP will be asserted with an error code. This error reporting mechanism is provided in cases where the 68020 unable to report errors. The assertion of these signals will notify the Host via the ESP that a Node Reset is requested. The Watchdog Timer can be programmed for up to 274 seconds before it expires if no Watchdog Counter Clear strobe is detected..

- **FDDI Chips CSR Bus Interface:** This is an interface to the ELM, MAC, MAC\_CAM, RMC and PARSER CSR Bus. The bus interface allows the MC68020 to perform read/write accesses of the Control/Status Registers contained within these units. A level of re-synchronization from the 68020's 64nsec clock to the FDDI Chips BYTCLK (80nsec) will occur such that the interface is in synch with the FDDI time domain. The CSR Bus is a 16-bit bi-directional data bus and will be justified to the 68020's "low word" or D31 thru D16. Word parity generation will be provided to the PARSER only.
- **ESP Interface:** This is an interface channel between the AMI Gate Array and the ESP Gate Array. This interface will provide access to the ESP/Host as well as PMC/ Packet Buffer Memory. The path to the Host is used for accessing the Host Command Ring and Unsolicited Ring. The path to the PMC is used for accessing PMC Registers and Packet Memory. Both paths are indirect methods of communications. The ESP Bus Interface supports generating and checking longword parity. Separate parity logic is contained within this section.
- **Interrupt Controller:** All the interrupts to the 68020 come into this function unit and are prioritized before connecting to the 68020. The 68020 has 7 IPLs ( Interrupt Priority Levels ), with level 7 being the highest level and non-maskable. Incoming interrupts will be synchronized to the 64nsec clock. Each interrupt will have associated with it a

unique vector, vector register contents will be programmable. Separate Interrupt Mask Bits settable via CSRs will be provided as a hardware mask for interrupts. Separate Interrupt Enable Bits settable via CSRs will be provided as individual hardware enables for interrupts.

**Table 1: INTERRUPT STRUCTURE**

| IPL | DESCRIPTION       | PRIORITIZED SOURCE                                                              |
|-----|-------------------|---------------------------------------------------------------------------------|
| 7   | System Error,RTOS | Watchdog Timer<br>AC Power Fail<br>AM/ESP Bus Parity Error (AMI Read operation) |
| 6   | XFA               | ESP Chip<br>FDDI CSR Bus Parity Error                                           |
| 5   | FDDI Interface    | ELM<br>PARSER                                                                   |
| 4   | FDDI Interface    | MAC<br>RES<br>RMC<br>SMT Priority                                               |
| 3   | XFA               | PM Start Transmit                                                               |
| 2   | System Required   | Hardware Timer                                                                  |
| 1   | RTOS Required     | Event Level Monitor                                                             |

- **Reset Logic:** The reset logic provides a path to reset the entire 68020 subsystem. Inputs to the Reset Logic are INIT\_L from the ESP G.A. and SRAM\_PE. The INIT\_L signal from the ESP G.A. can be asserted due to: Power Cycle, Backplane Reset or Node Reset ( Device Driver Generated ). The manner in which the Watchdog Timer resets the Adapter Manager is by requesting a Node Reset from the Host via the error bits XERCOD[1:0]H that are a direct connect to ESP.

The XMI signal AC\_LO L will be used by a portion of this logic to indicate the difference between a Cold and Warm Reset. AC\_LO L does not get asserted on a Warm Reset, but only on a cold power-up Reset. The Parity Error Address Register contents are maintained through a Warm Reset. The Parity Error Address Register is only initialized on a Cold Reset by a signal derived from AC\_LO L.

When an SRAM Parity Error occurs, logic will put the 68020 into a Reset State and not remove it until INIT\_L cycles.

A Reset Disable Bit is provided in the GPCSR to prevent the INITL line from resetting the AMI during a warm reset. This allows the 68020 to generate a reset of the adapter without losing its code. The bit is automatically cleared after the INITL line is cycled once.

- **Clock Logic & Drivers:** Logic is provided to create a 50% duty cycle 68020 Clock from the 33% Duty Cycle CLK34H. Internal clocks are buffered and their loading is balanced as much as possible to minimize clock skew throughout the Gate Array.

- **Parity Logic:** This logic generates and checks SRAM Data Bus Parity. The Byte Parity generated defaults to ODD Parity. Provisions are made to generate EVEN Parity to facilitate testing of the Parity circuit. A parity error indication is obtained only on an SRAM read cycle.

## 1.4 68020 MEMORY SUBSYSTEM

Contained in this subsystem will be the local memory for the MC68020. The subsystem would include: SRAM, EEPROM and the Default Physical Address PROM. Buffered versions of the 68020 address and data bus will be connected to the Memory Subsystem data path to provide added drive capability for interfacing to the memory array. The address and lines to the Memory Subsystem will be buffered externally from the AMI Gate Array providing added drive capability.

- **SRAM:** The SRAM will be used for operand loads and stores as well as program space during program execution. The SRAM will be configured to provide a density of 256KByte of zero wait state memory. Byte Parity will be provided with 4 bits of Parity Protection for the SRAM data bus.
- **EEPROM:** EEPROM will be provided to contain code for: Kernel Code, Self Test, Initialization, RTOS Monitor, SMT, CMT, Port Control, XID/TEST, Local and Remote Management, etc. The contents of EEPROM will be transferred over to SRAM after diagnostics have been run on the SRAM. This transfer of memory contents will be performed by the 68020. This will allow code to run with zero wait states.

EEPROM will be configured to provide a density of 256KByte and will be accessed via longwords only. Each longword read from EEPROM requires 4 wait states due to the access time of the EEPROM ( 250nsec ). A special software timed operation is required when writing to EEPROM due to the special nature of the devices. A 1 microsecond delay must be inserted between page mode writes and a 10 millisecond delay must be inserted between different page write accesses while contents are being moved within the EEPROMs.

- **DEFAULT PHYSICAL ADDRESS PROM:** The Default Physical Address ( Node Address ) PROM will contain six bytes of FDDI Node Address unique to each adapter. The 68020 will have ability to read the entire 32 Bytes within the PROM. DPA PROM data integrity will be protected by Checksum.

## 1.5 SUBSYSTEM DATA INTEGRITY

Parity will be used to provide SRAM Data Integrity Protection as well as AMI/ESP Bus Protection. EEPROM contents integrity will be confirmed with several Checksum calculations and compares. Separate checksums will be calculated for each major block of code, such as Self Test Code, Functional Code, etc. Byte Parity will be provided on the SRAM Data Bus and Longword Parity will be provided on all transactions to the AMI/ESP Bus. Word Parity will be generated for writes to the FDDI CSR Bus on accesses to the Parser only.

## 1.6 POWER UP & SELF TEST

- **Power-Up Reset:** The ESP Chip will provide an INIT\_L Signal to the AMI. Upon de-assertion of the Reset Signal, the 68020 will obtain its Stack Pointer from Address \$00000000 and its Program Counter from Address \$00000004 in EEPROM and begin executing Start Up and Self Test Code.

Test Logic within the AMI will be provided only as needed to ease the task of self test. The AMI will rely mainly on the 68020 performing the testing of the AMI Gate Array. The 68020 will run Self Test Code out of EEPROM until it has sufficiently tested enough SRAM to enable it to safely run Self Test Code out of SRAM. Self Test Code in EEPROM will be copied over to SRAM by the 68020. Executing code out of SRAM will minimize self test execution time. Self Test execution time is critical due to the 10 second maximum time imposed by the XMI Bus.

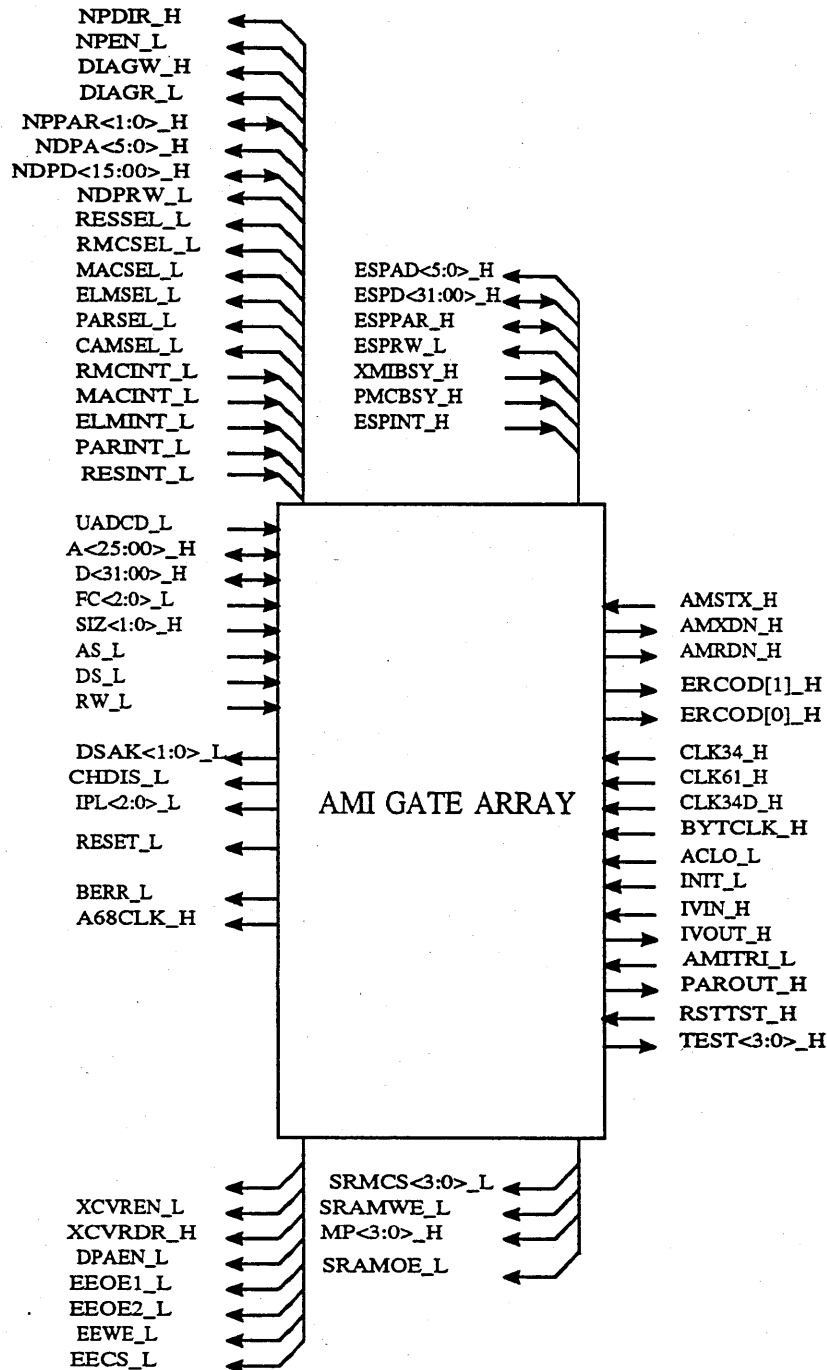
- **Error Status:** Two dedicated lines from the AMI will go to the ESP and PMC Gate Arrays. These lines, XERCOD[1:0], will provide 4 possible codes that represent the error status of the AMI. The status conditions are: Null, Watchdog Timer Expired, SRAM Parity Error and ESP Bus Parity Error.
- **Loading Functional Code:** Once Self Test is complete, the 68020 will move the Functional Code from EEPROM to SRAM. Upon completion of this, the 68020 will execute other initialization procedures to transition to Uninitialized State and await a command from the Host.



# CHAPTER 2

## AMI CONTEXT

Figure 3: AMI CONTEXT



## SIGNAL PINS ON AMI GATE ARRAY

### 2.1 68020 INTERFACE

The following signal pins on the 68020 microprocessor will have a connection to the AMI Gate Array.

- **A<25:00>\_H**: This is the 26-Bit Address Bus used to address any of 67,108,864 bytes.
- **D<31:00>\_H**: This is the 32-Bit Data Bus used to transfer 8, 16, 24 or 32 bits of data per bus cycle.
- **FC<2:0>\_L**: This 3-Bit Function Code is used to identify the address space for each bus cycle.
- **SIZ<1:0>\_H**: Size indicates the number of bytes remaining to be transferred for this cycle. These signals, together with A0 and A1, define the active sections of the data bus.
- **AS\_L**: Address Strobe indicates that a valid address is on the bus.
- **DS\_L**: Data Strobe indicates that valid data is to be placed on the data bus by an external device or has been placed on the data bus by the MC68020.
- **RW\_L**: Read/Write defines the bus transfer as an MPU read or write.
- **DSAK<1:0>\_L**: Data Transfer and Size Acknowledge are bus response signals that indicate the requested data transfer operation is completed. In addition, these two lines indicate the size of the external bus port on a cycle-by-cycle basis.
- **CHDIS\_L**: Cache Disable disables the on-chip cache to assist emulator support.
- **IPL<2:0>\_L**: Interrupt Priority Level provides an encoded interrupt level to the processor.
- **RESET\_L**: Reset signal to the 68020.
- **BERR\_L**: Bus Error indicates that an invalid or illegal bus operation is being attempted.
- **A68CLK\_L**: Clock signal driven to input of 68020 microprocessor.

### 2.2 LOCAL MEMORY INTERFACE

These signals connect to the 68020's Memory Subsystem

- **XCVREN\_L**: This control signal enables the Memory Subsystem bidirectional transceivers.
- **XCVRDR\_H**: This determines the direction of data flow through the Memory Subsystem 74F657 bidirectional transceivers.
- **DPAEN\_L**: This enable signal when asserted along with a stable address, enables the contents of the DPA ROM onto the 68020 local data bus.
- **EEOE1\_L**: EEPROM BANK 1 Output Enable signal controls the EERPOM data output buffers and is used to initiate read operations. This signal must be asserted for all Read operations from the first bank of EEPROM's.
- **EEOE2\_L**: EEPROM BANK 2 Output Enable signal controls the EERPOM data output buffers and is used to initiate read operations. This signal must be asserted for all Read operations from the second bank of EEPROM's.
- **EEWE\_L**: The Write Enable signal controls the writing of data to the EEPROM's.

- **EECS\_L**: The EEPROM Chip Select signal controls the selection of the EEPROMs on both read and write cycles. A further level of Bank Selection is done in the Module PAL prior to connecting to the EEPROMs.
- **SRMCS<3:0>\_L**: Static RAM Chip Selects for 4 individual byte groupings. SRAM Bank select is provided by A<17>\_H and A<17>\_L connected to the different Bank alternate Chip Select Lines ( CE2 ).
- **SRAMWE\_L**: Static RAM Write Enable.
- **SRAMOE\_L**: Static RAM Output Enable.
- **MP<3:0>\_H**: These are the parity bits used during each data cycle for parity checking and generation of the SRAM Data Bus.

### 2.3 ESP INTERFACE

These signals connect to the ESP Gate Array.

- **ESPAD<5:0>\_H**: These address line provide 64 possible directly addressed longword registers within the ESP Gate Array.
- **ESPD<31:00>\_H**: This is the data bus connecting the ESP and the AMI. It is a bidirectional tri-stateable CMOS data bus.
- **ESPPAR\_H**: Longword Parity is generated and checked on the ESP Bus via this parity bit.
- **ESPRW\_L**: The Read/Write Line defines the direction of the data transfer.
- **XMBSY\_H**: This signal asserted indicates to the AMI that the last transaction that was requested thru the ESP Interface intended for the XMI Host has not completed yet and is still in-process. The next transaction intended for the XMI should not start until the BUSY Bit is de-asserted.
- **PMCBSY\_H**: This signal asserted indicates to the AMI that the last transaction that was requested thru the ESP Interface intended for the PMC has not completed yet and is still in-process. The next transaction intended for the PMC should not start until the BUSY Bit is de-asserted.
- **ESPINT\_H**: ESP to AMI Interrupt. Service routine in the 68020 must determine the reason for the interrupt by reading the ESP interrupt status registers. Error conditions within ESP and PMC will be identified thru this interrupt mechanism as well.

### 2.4 PMC SIGNALS

These signals connect directly between the PMC Gate Array and the AMI Gate Array.

- **AMSTX\_H**: Adapter Manager Start Transmit Interrupt is an interrupt to the 68020 indicating a packet is available for the Adapter Manager. During selftest this line is used as a test pin for DRAM refresh.
- **AMXDN\_H**: Adapter Manager Transmit Done is a signal that tells the PMC Gate Array that the 68020 has taken a packet off the Adapter Manager's Transmit Ring. This signal is asserted after every packet transaction.

- **AMRDN\_H:** Adapter Manager Receive Done is a signal that tells the PMC Gate Array that the 68020 has put a packet onto the Adapter Manager's Receive Ring. This signal is asserted after every packet transaction.

## 2.5 CSR BUS INTERFACE

These signals connect to the FDDI Chips CSR Bus.

- **BYTCLK\_H:** This is the 80nsec, 50% duty cycle clock used by the FDDI Chip Set. This is used for synchronization of the CSR Bus Interface with the AMI Gate Array.
- **NDPA<5:0>\_H:** These address lines provide 64 possible directly addressed word locations within an FDDI CHIP on the CSR BUS. Note that all FDDI chips do not contain 64 word locations.
- **NDPD<15:00>\_H:** This is a bidirectional three-state data bus used to exchange data between the FDDI chips and the AMI Gate Array.
- **NDPRW\_L:** This Read/Write output indicates, to the FDDI chips, whether the current bus cycle is a read (NPRW = 1) or a write (NPRW = 0) CSR cycle.
- **RMSEL\_L:** This output selects the RMC FDDI chip for the current bus cycle.
- **MACSEL\_L:** This output selects the MAC FDDI chip for the current bus cycle.
- **ELMSEL\_L:** This output selects the ELM FDDI chip for the current bus cycle.
- **PARSEL\_L:** This output selects the PARSER Gate Array for the current bus cycle.
- **CAMSEL\_L:** This output selects the CAM chip for the current bus cycle.
- **RESSEL\_L:** This output select is reserved for selecting the Encryption Chip or another device on the FDDI CSR Bus in future products.
- **RMCINT\_L:** This input indicates an interrupt request from the RMC FDDI chip.
- **MACINT\_L:** This input indicates an interrupt request from the MAC FDDI chip.
- **ELMINT\_L:** This input indicates an interrupt request from the ELM FDDI chip.
- **PARINT\_L:** This input indicates an interrupt request from the PARSER Gate Array.
- **RESINT\_L:** This input is reserved for the interrupt request from the Encryption Chip or another device on the FDDI CSR Bus in future products.
- **DIAGW\_H:** This output is a write strobe to the adapter's Diagnostics Registers.
- **DIAGR\_L:** This output is an enable line for the adapter's Diagnostics Buffer Read Back.
- **NPDIR\_H:** This output bit is the direction control for the FDDI CSR Bus Transceiver chips.
- **NPEN\_L:** This output is the enable for the FDDI CSR Bus Transceiver chips.

## 2.6 ATE TEST SIGNALS

- **IVIN\_H:** Signal used for Gate Array testing.
- **IVOUT\_H:** Signal used for Gate Array testing.
- **PAROUT\_H:** Signal used for Gate Array testing.
- **AMITRI\_L:** AMI Tristate signal is used to tristate all outputs for Gate Array test.

- **RSTTST\_H:** Reset Test is a bit that controls the mux select that provides the count value for the Reset Counter. For simulation purposes, when this bit is set "high" the Reset Counter will count 7 clocks and release the RESETL and allow the 68020 to wake up. When this bit is "low" the Reset Counter will count for 101msec before releasing RESETL and allowing the 68020 to wake up.
- **TEST<3:0>\_H:** These test bits are outputs from the FDDI\_CSR\_BUS Interface State Machine or the ESP Interface Control Logic. The output is selected by Bit<12> in the GPCSR "low" = ESP Interface, "high" = FDDI\_CSR\_BUS Interface.

## 2.7 RESET INTERFACE

- **INIT\_L:** This is a reset signal provided by the ESP's Reset Logic. It is de-asserted when the adapter power supplies reach a stable state after power-up. This signal can also be cycled due to a host software generated Node Reset as well as by a ESP register write by the 68020.
- **ACLO\_L:** ACLO\_L indicates that AC Power to the system power supply has dropped below minimum specification. This signal is an early warning indicator that DC Power will fail in 4 msec. This signal will be used as a nonmaskable interrupt to the 68020.

## 2.8 XMI CORNER INTERFACE

- **CLK34\_H:** This is one phase of the 6-phase clock provided by the XMI Corner. The Clock is 33% duty cycle with a 64nsec period. This clock is "high" for 21.3nsec and low for 42.7nsec.
- **CLK34D\_H:** This is a 10.7nsec delayed version of XMI\_XCI\_34H clock that is required to assist in the generation of a 50% duty cycle 68020 Clock. This clock is "high" for 21.3nsec and low for 42.7nsec.
- **CLK61\_H:** This is another phase of the 6-phase clock provided by the XMI Corner. The Clock is 33% duty cycle with a 64nsec period. This clock is "high" for 21.3nsec and low for 42.7nsec. This clock is delayed from CLK34\_H by 32nsec.

## 2.9 AMI SYSTEM INTERFACE

- **UADCD\_L:** Upper Address Decode is used to inform the AMI Gate Array when A<31:24> are all ones. This signal comes from an external PAL.
- **ERCOD<1:0>:** These signals indicates to the ESP and XMI that a severe error condition exists that the 68020 may not be able to report.

# CHAPTER 3

## ADDRESS DECODE LOGIC

### 3.1 OVERVIEW

This logic monitors the 26 address bits from the 68020 and the pre-decoded PAL output for address lines A<31:D26>. The 68020's physical address space is decoded to generate the select signals for all logic accessed by the 68020. The AMI receives only 26 address lines from the 68020 enabling an address space of up to 64MBytes.

Also decoded in this section are invalid accesses to Nonexistent Memory and unsupported CPU Space Cycles: Access Level Control, Coprocessor Communication.

Figure 4: ADDRESS DECODER - BLOCK DIAGRAM

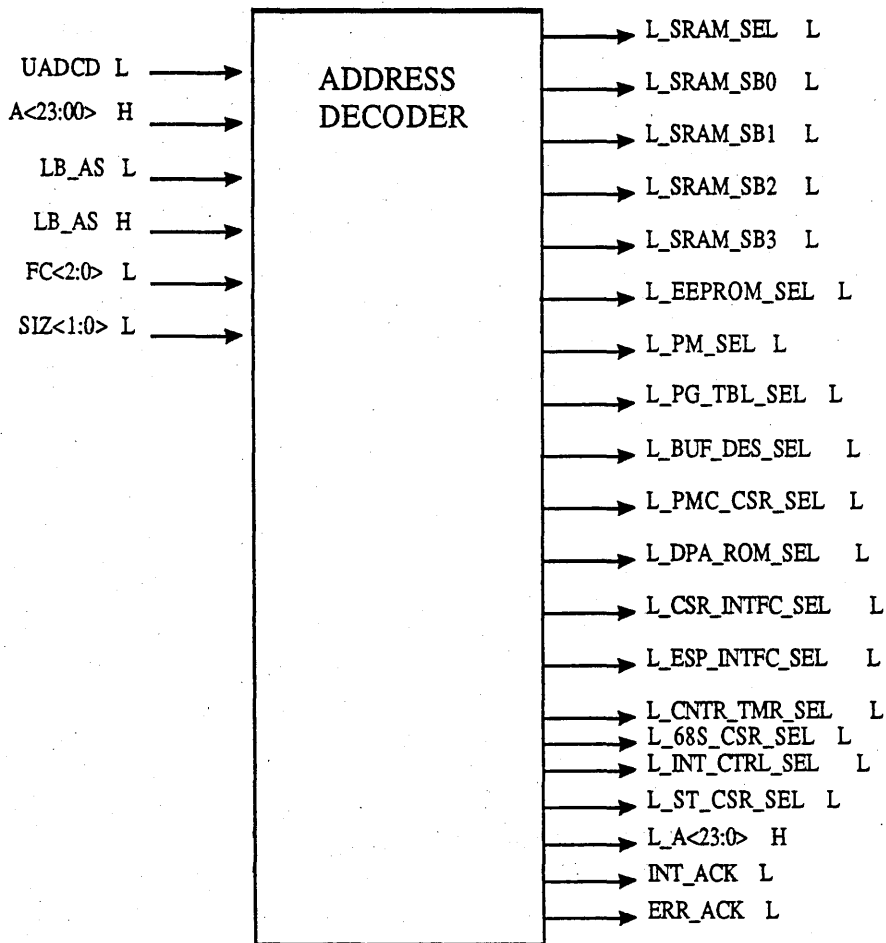
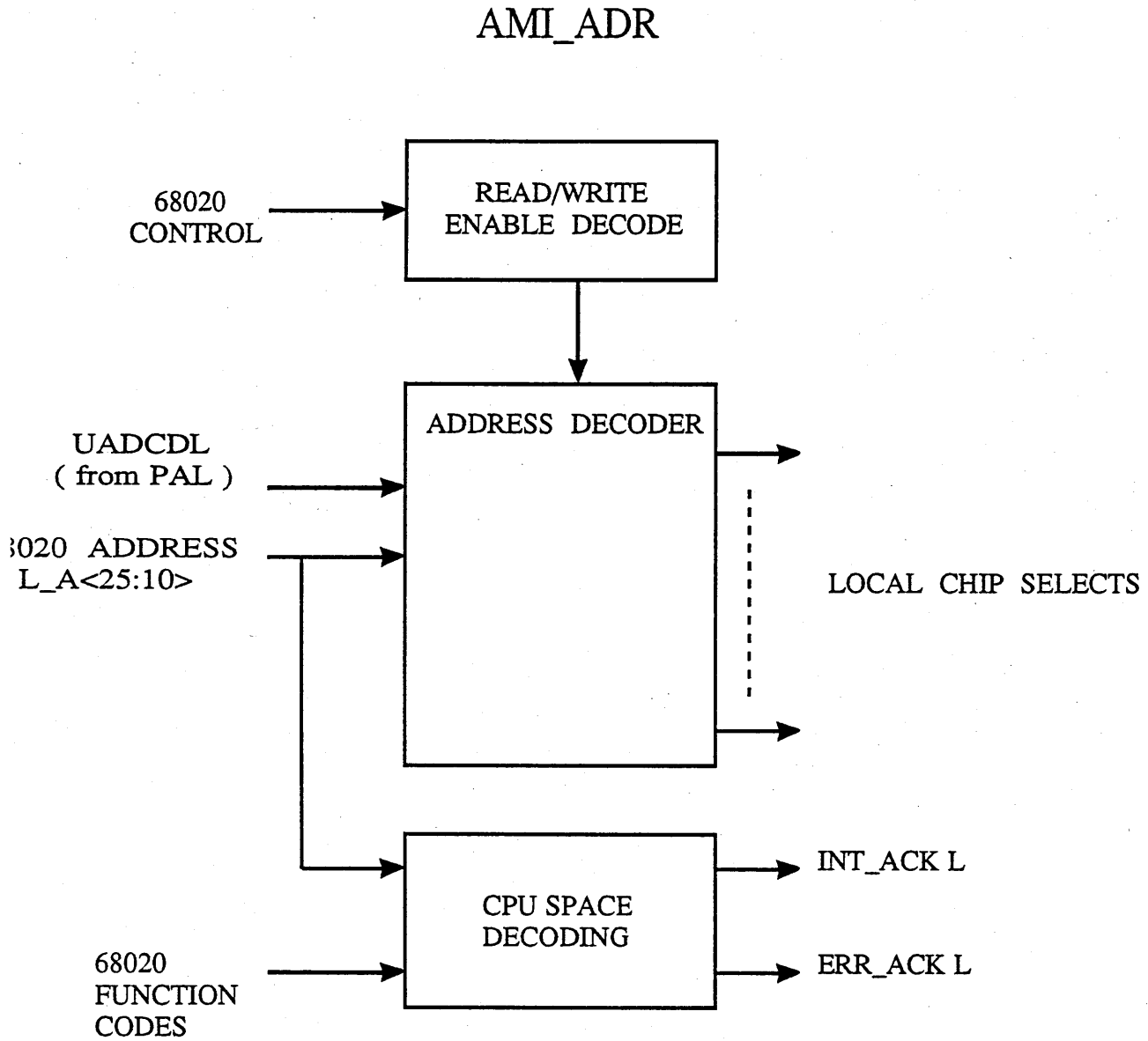


Figure 5: AMI\_ADR - 2ND LEVEL BLOCK DIAGRAM



### 3.2 68020 INTERFACE

- **UADCD** L Upper Address Decode for 68020 address lines A<31:24> H. The decode is done by a 16L8D.
- **A<25:0>** H 68020 Address Bus.
- **FC<2:0>** H 68020 Function Codes. These codes identify the processor state and the address space of the bus cycle currently being executed.
- **SIZ<1:0>** H 68020 Transfer Size.

### 3.3 LOCAL G.A. SIGNALS

- **L\_AS** L Local version of 68020 Address Strobe signal - logic "low" true.
- **L\_AS** H Inverted local version of 68020 Address Strobe signal - logic "high" true.
- **L\_SRAM\_SEL** L Local SRAM Select line - qualified by L\_AS L.
- **L\_SRAM\_SB0** L Local SRAM Select for Byte 0 - qualified by L\_SRAM\_SEL L.
- **L\_SRAM\_SB1** L Local SRAM Select for Byte 1 - qualified by L\_SRAM\_SEL L.
- **L\_SRAM\_SB2** L Local SRAM Select for Byte 2 - qualified by L\_SRAM\_SEL L.
- **L\_SRAM\_SB3** L Local SRAM Select for Byte 3 - qualified by L\_SRAM\_SEL L.
- **L\_EEPROM\_SEL** L Local EEPROM Chip Select. Supports longwords accesses only.
- **L\_PM\_SEL** L Local Packet Memory Select - supports direct address mode in the AMI/ESP Interface.
- **L\_PG\_TBL\_SEL** L Local Page Table Select - supports direct address mode in the AMI/ESP Interface.
- **L\_BUF\_DES\_SEL** L Local Buffer Descriptor Select - supports direct address mode in the AMI/ESP Interface.
- **L\_PMC\_CSR\_SEL** L Local PMC CSR Select - supports direct address mode in the AMI/ESP Interface.
- **L\_DPA\_ROM\_SEL** L Device Physical Address ROM ( Node Address ) Chip Select. Supports byte wide operations only. Makes use of Motorola Byte 0 on D[31:24].
- **L\_CSR\_INTFC\_SEL** L CSR Interface select line for word wide ( 16-bit ) data transfers.
- **L\_ESP\_INTFC\_SEL** L AMI/ESP Bus Interface select line for longword register accesses to the ESP Chip.
- **L\_CNTR\_TMR\_SEL** L Counter/Timer logic select line for longword accesses.
- **L\_INT\_CTRL\_SEL** L Interrupt Controller logic select line for word wide ( 16-bit ) data transfers.
- **L\_68S\_CSR\_SEL\_L** General AMI Gate Array CSR registers select line.
- **L\_ST\_CSR\_SEL\_L** Self Test Logic select line.
- **L\_A<25:0>** H Local buffered version of the 68020 Address Bus four use by AMI Sub-sections.
- **INT\_ACK** L High Level Interrupt Acknowledge cycle decode signal.



- **ERR\_ACK L** Error Condition signal asserted when 68020 executing in CPU Space Cycles: Access Level Control or Coprocessor Communication Cycle is executed. These cycles are decoded off of the CPU Space Field ( A19-A16 and FC<2:0> ). The intent is for a Bus Error ( BERR\_L ) to be generated in the event that one of the above stated cycles is executed.

Figure 6: ADDRESS MAP - BLOCK DIAGRAM

|             |                               |        |
|-------------|-------------------------------|--------|
| \$FFFF FFFF | NOT DECODED                   | < 16GB |
| \$0400 0000 | BUFFER DESCRIPTOR SPACE       | 16MB   |
| \$0300 0000 | PAGE - TABLE SPACE            | 16MB   |
| \$0200 0000 | PACKET MEMORY SPACE           | 16MB   |
| \$0100 0000 | NOT DECODED                   | 13MB   |
| \$0030 0000 | RESERVED FOR SRAM EXPANSION   | 768kB  |
| \$0024 0000 | SRAM                          | 256KB  |
| \$0020 0000 | RESERVED                      | < 1MB  |
| \$0010 2000 | PMC CSR SPACE                 | 1KB    |
| \$0010 1C00 | SELF TEST SPACE               | 1KB    |
| \$0010 1800 | INTERRUPT CONTROLLER          | 1KB    |
| \$0010 1400 | 68S CSRs                      | 1KB    |
| \$0010 1000 | COUNTER - TIMERS              | 1KB    |
| \$0010 0C00 | ESP BUS INTERFACE             | 1KB    |
| \$0010 0800 | FDDI CSR BUS INTERFACE        | 1KB    |
| \$0010 0400 | DPA ROM                       | 1KB    |
| \$0010 0000 | RESERVED FOR EEPROM EXPANSION | 768KB  |
| \$0004 0000 | EEPROM                        | 256KB  |
| \$0000 0000 |                               |        |

# CHAPTER 4

## AMI CONTROL LOGIC

### 4.1 OVERVIEW

This logic works in conjunction with the Address Decode Logic to generate a Data Transfer & Size Acknowledge signal to the 68020. The DSACK<1:0> L signals indicate that a data transfer is complete and the port size of the external device (8-,16-, 32-bits). During a read cycle, when the processor recognizes DSACK<sub>x</sub>, it latches the data and then terminates the bus cycle; during a write cycle, when the processor recognizes DSACK<sub>x</sub>, the bus cycle is terminated.

During an operand transfer cycle, the slave device signals its port size (byte, word or long-word) and transfer status (complete or not complete) to the 68020 through the DSACK<1:0> lines. The logic in this section decodes the address and determines the width of the port for the current transfer. A wait state generator is included for slow peripherals such as the EEPROMs.

A Bus Error Timer is included in this function block. It is implemented with a 16-bit down counter that should never expire under normal conditions. The counter is programmable and can have a maximum time before issuing a BERR L of 4msec. The counter is enabled by the beginning of a valid 68020 Bus Cycle. Under normal operation it is reset by the occurrence of any DSACK<1:0>\_L. The counter is set to value longer than the longest valid bus cycle. If there is an access to Non-existent Memory Space, no DSACK will be generated and the timer will expire. When the timer expires, it generates a Bus Error asserting BERR\_L.

This block also decodes the 68020 control signals and generate the necessary local control signals to perform read and write cycles to registers within the Gate Array. Separate logic will exist within the Memory Subsystem Control block to provide the control signals to the Memory Subsystem.

Figure 7: AMI CONTROL LOGIC - BLOCK DIAGRAM

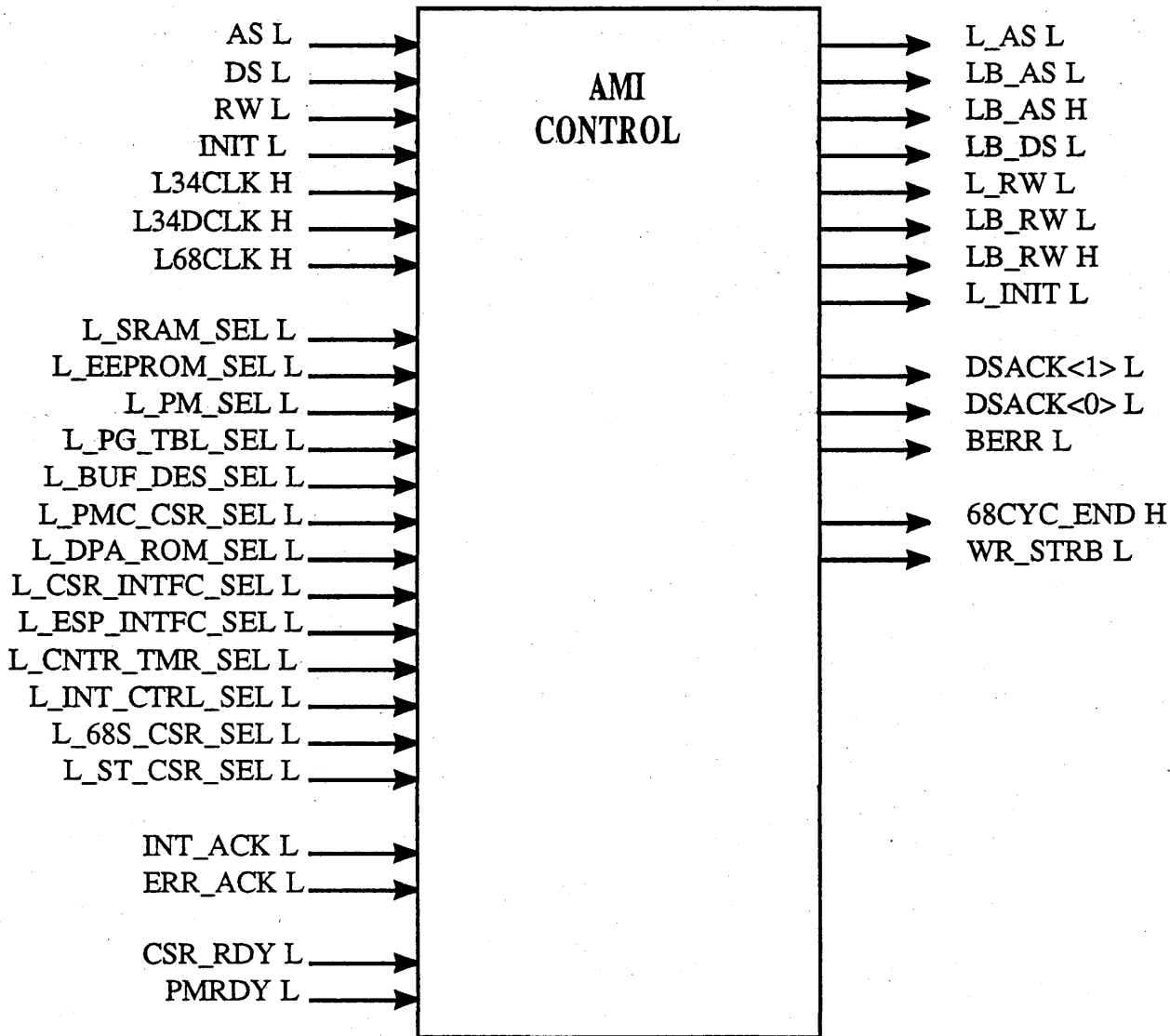
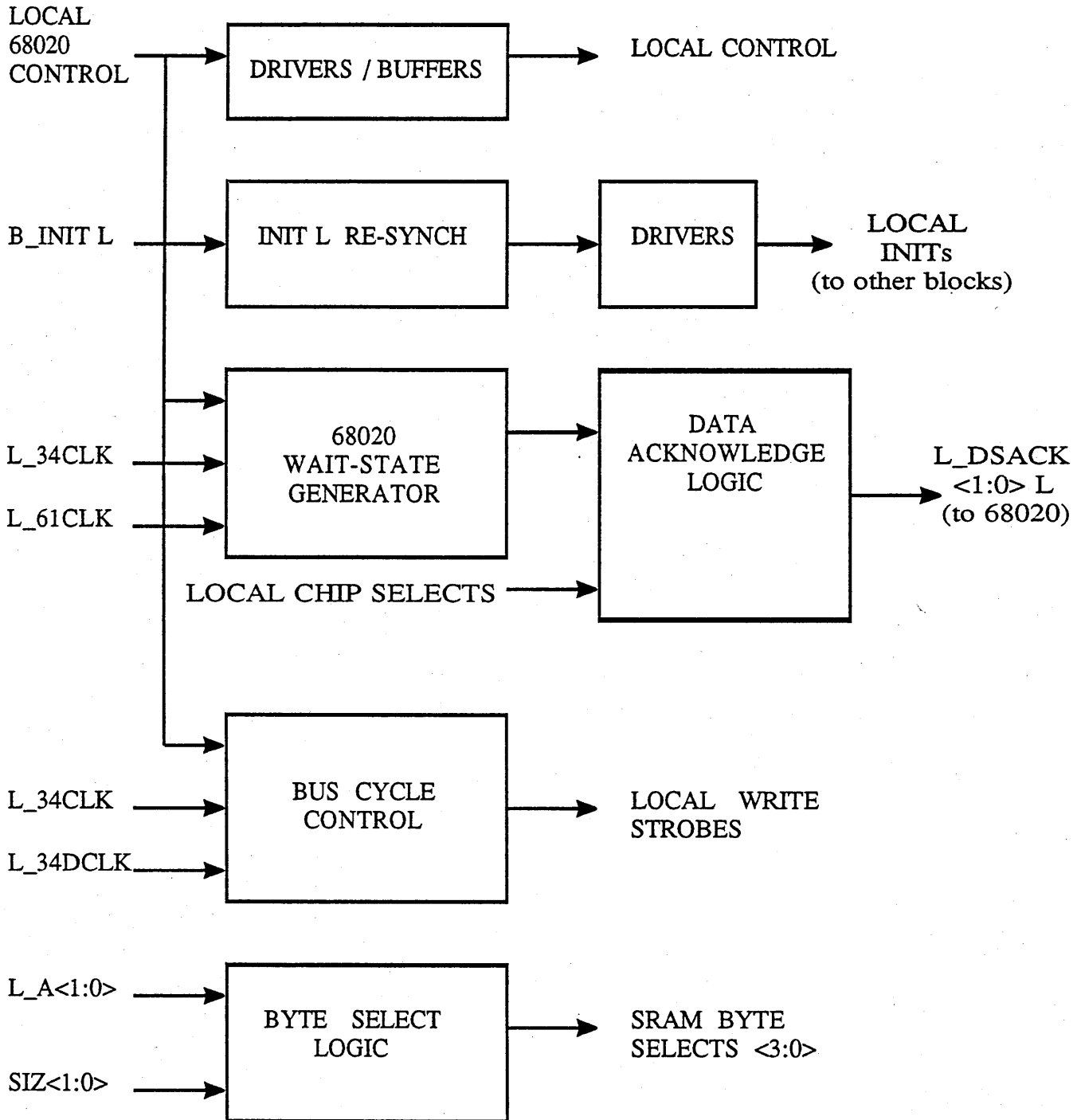


Figure 8: AMI\_CTRL - 2ND LEVEL BLOCK DIAGRAM

# AMI\_CTRL



## 4.2 LOCAL INPUTS

- **AS L 68020 Address Strobe.**
- **DS L 68020 Data Strobe.**
- **RW L 68020 Read/Write Bus Direction Control.**
- **INIT L Initialize ( Reset ) Signal from the ESP Chip.**
- **L34CLK H Local Buffered version of XMI Clock XCI 34H. It has a 64nsec period.**
- **L34DCLK H Local Buffered version of 10.7nsec delayed XMI Clock XCI 34H. It has a 64nsec period.**
- **L68CLK H Local Buffered version of the 50% duty cycle 68020 Clock with a 64nsec period.**
- **L\_SRAM\_SEL L SRAM Select line.**
- **L\_EEPROM\_SEL L Local EEPROM Chip Select. Supports longwords accesses only.**
- **L\_PM\_SEL L Local Packet Memory Select - supports direct address mode in the AMI/ESP Interface.**
- **L\_PG\_TBL\_SEL L Local Page Table Select - supports direct address mode in the AMI/ESP Interface.**
- **L\_BUF\_DES\_SEL L Local Buffer Descriptor Select - supports direct address mode in the AMI/ESP Interface.**
- **L\_PMC\_CSR\_SEL L Local PMC CSR Select - supports direct address mode in the AMI/ESP Interface.**
- **L\_DPA\_ROM\_SEL L Device Physical Address ROM ( Node Address ) Chip Select. Supports byte wide operations only. Makes use of Motorola Byte 0 on D[31:24].**
- **L\_CSR\_INTFC\_SEL L CSR Interface select line for word wide ( 16-bit ) data transfers.**
- **L\_ESP\_INTFC\_SEL L AMI/ESP Bus Interface select line for longword register accesses to the ESP Chip.**
- **L\_CNTR\_TMR\_SEL L Counter/Timer logic select line for longword accesses.**
- **L\_INT\_CTRL\_SEL L Interrupt Controller logic select line for word wide ( 16-bit ) data transfers.**
- **L\_68S\_CSR\_SEL\_L General AMI Gate Array CSR registers select line.**
- **L\_ST\_CSR\_SEL\_L Self Test Logic select line.**
- **INT\_ACK L High Level Interrupt Acknowledge cycle decode signal.**
- **ERR\_ACK L Error Condition signal asserted when 68020 executing in CPU Space Cycles: Breakpoint Acknowledge, Access Level Control or Coprocessor Communication Cycle is executed. These cycles are decoded off of the CPU Space Field ( A19-A16 and FC<2:0> ). The intent is for a Bus Error ( BERR\_L ) to be generated in the event that one of the above stated cycles is executed.**
- **CSR\_RDY L This signal notifies the DSACK Logic that the CSR Interface has acknowledged the data cycle from the 68020. This signal must be provided by the interface since synchronization occurs in the CSR interface.**

- **PMRDY L** This signal notifies the DSACK Logic that the AMI/ESP Interface has acknowledged the data cycle from the 68020.

### 4.3 LOCAL OUTPUTS

- **L\_AS L** Local version of 68020 Address Strobe - logic "low" true.
- **LB\_AS L** Local buffered version of 68020 Address Strobe - logic "low" true.
- **LB\_AS H** Local buffered version of 68020 Address Strobe - logic "high" true.
- **LB\_DS L** Local buffered version of 68020 Data Strobe - logic "low" true.
- **L\_RW L** Local version of 68020 Read/Write bus direction control.
- **LB\_RW L** Local buffered version of 68020 Read/Write bus direction control.
- **LB\_RW H** Local inverted buffered version of 68020 Read/Write bus direction control.
- **L\_INIT L** Local version of the XFA Adapter INIT L - used to indicate logic reset condition.
- **68CYC\_END H** This is a local control signal used to indicate the end of a 68020 bus cycle. Used in the Memory Control Logic.
- **WR\_STRB\_L** This signal is a retimed version of the 68020 Data Strobe used for all register accesses within the Gate Array.

### 4.4 68020 CONNECTIONS

- **DSACK<1:0> L** Data Transfer Acknowledge Signals to the 68020. These signals indicate that a data transfer is complete and the port size of the addressed device.

| DSACK1_L | DSACK0_L | RESULT                                         |
|----------|----------|------------------------------------------------|
| 1        | 1        | Insert Wait-States in current cycle            |
| 1        | 0        | Complete cycle - Data Bus Port Size is 8 Bits  |
| 0        | 1        | Complete cycle - Data Bus Port Size is 16 Bits |
| 0        | 0        | Complete cycle - Data Bus Port Size is 32 Bits |

- **BERR L** Bus Error signal to the 68020. Informs the processor that there has been a problem with the the bus cycle currently being executed. The problems may be:
  - Non-responding devices or non-existent memory.
  - Interrupt vector number acquisition failure.
  - Various other application dependent errors.
- **BERR\_L** interacts with the **HALT\_L** signal to determine if the current bus cycle should be re-run or aborted with a bus error. This implementation does not support bus cycle retries.

# CHAPTER 5

## RTOS, BUS ERROR & WATCHDOG - TIMERS

### 5.1 OVERVIEW

This block provides the Watchdog Timer the Bus Error Timer and the RTOS\_TICK Timer required by the RTOS Operating System.

All timers will have their count loaded by the 68020 at initialization. Timer outputs may be polled or directed to a specified interrupt level. Timers will be run off of the 64nsec clock that is derived from the XCI\_34H XMI Clock. Registers are provided for loading the Watchdog, Bus Error and RTOS Counters. A CSR is provide to enable and disable the timers as well as setting up self test modes of operation. CSR bits are provided to allow chaining of counter stages to reduce Self Test time.

Figure 9: COUNTER/TIMER

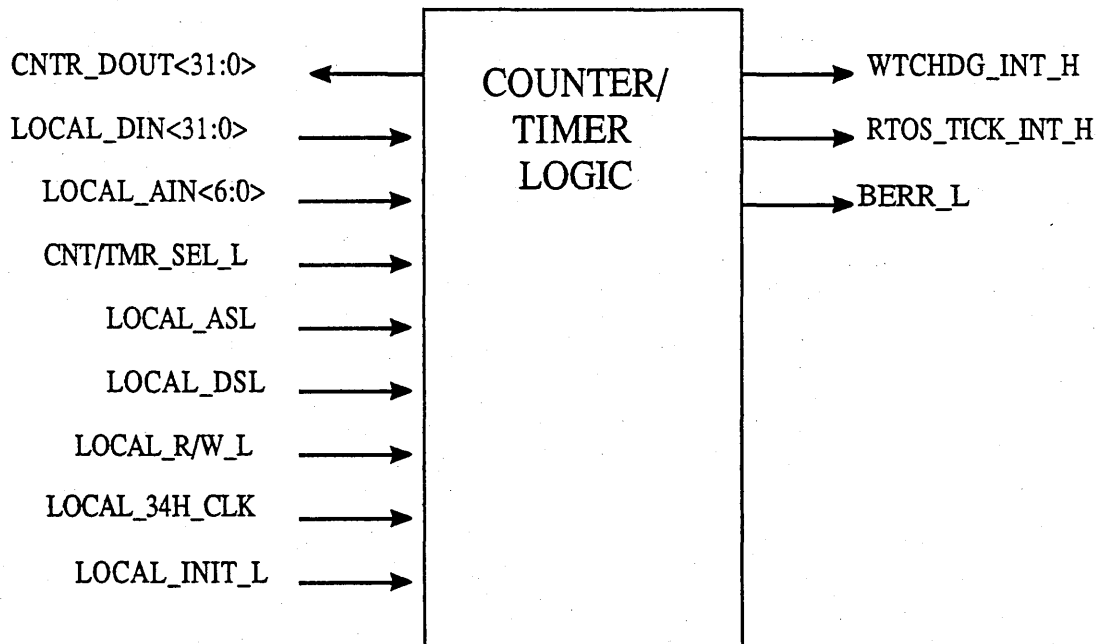




Figure 10: AMI\_TMRS - 2ND LEVEL BLOCK DIAGRAM

### AMI\_TMRS

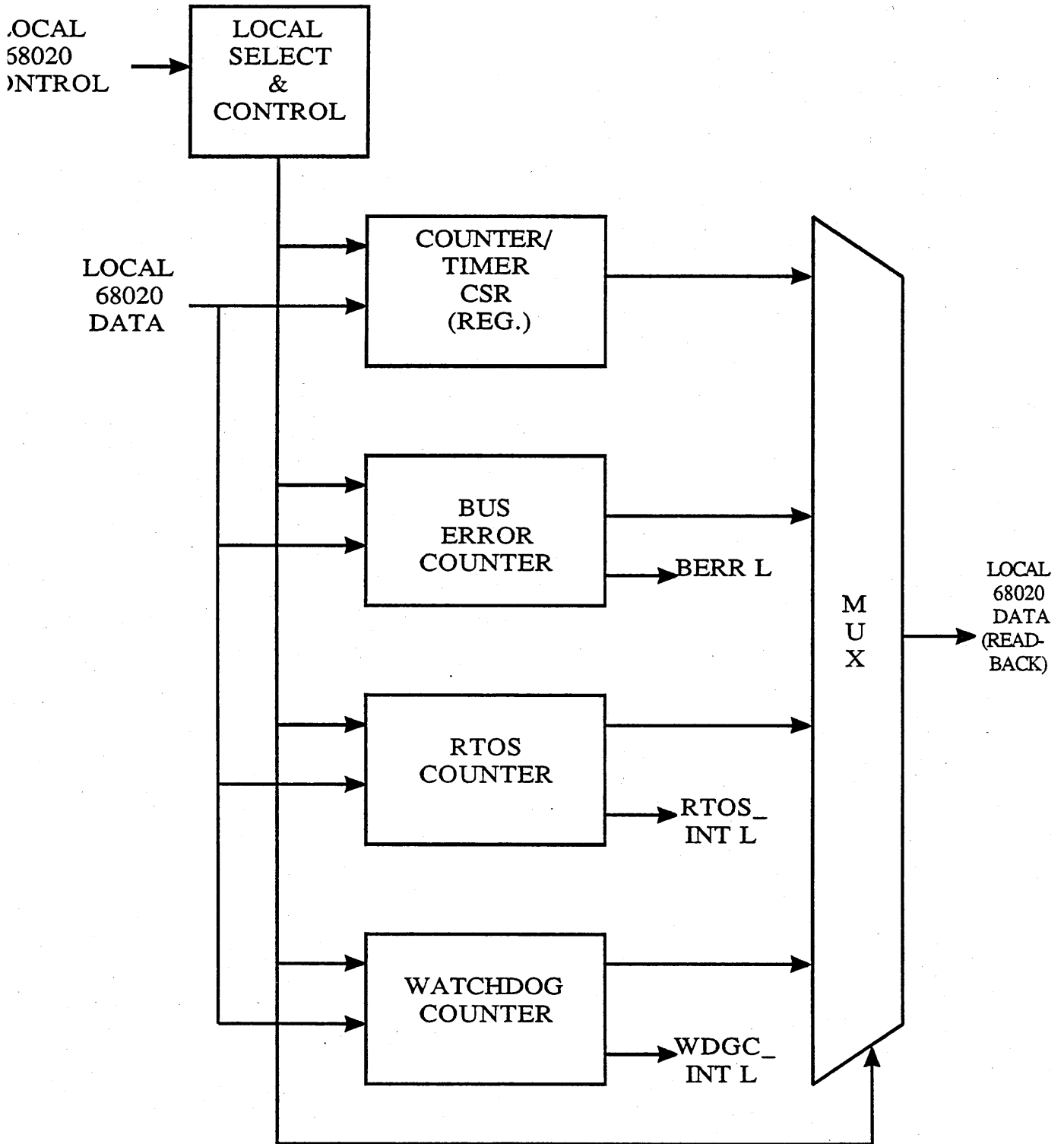
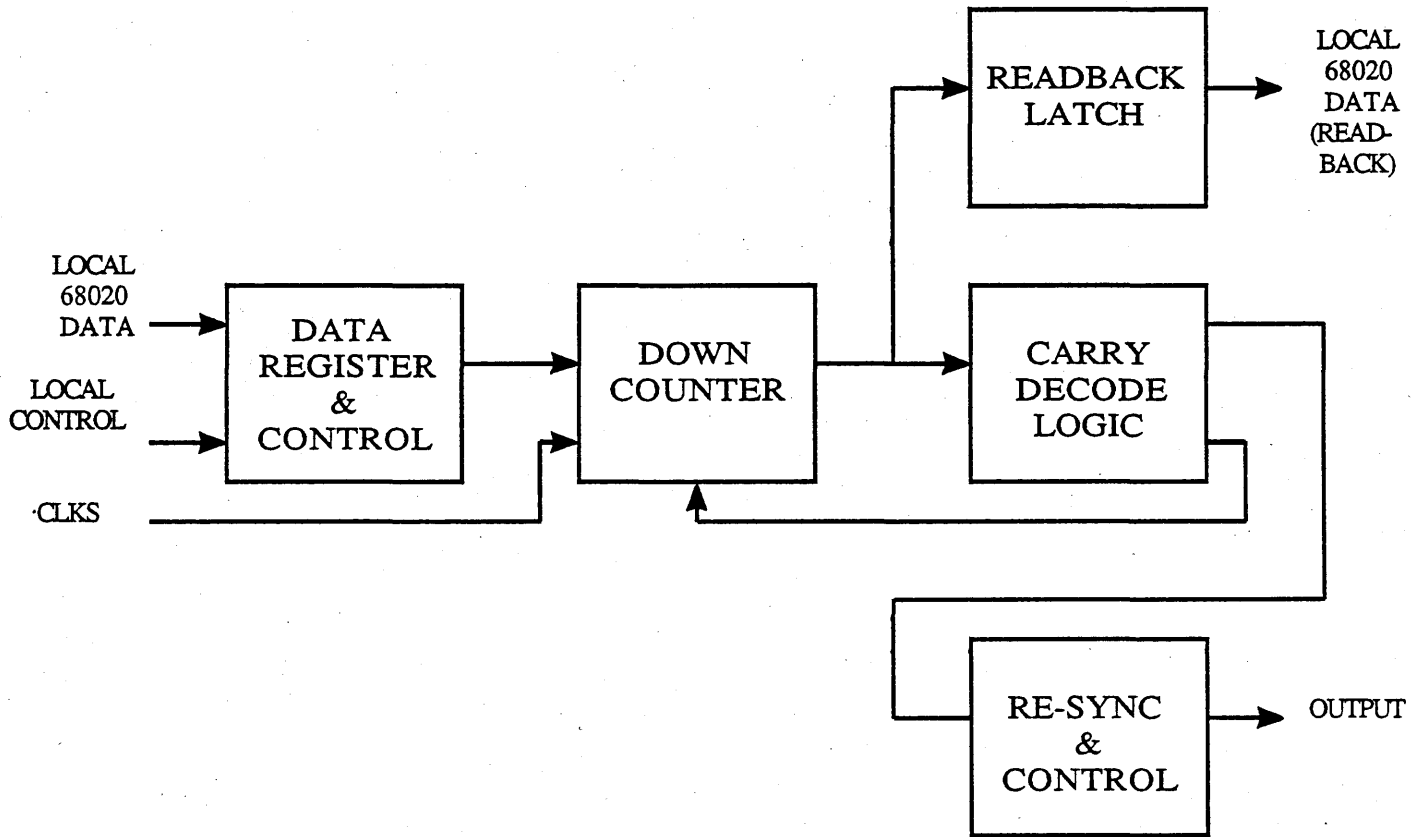


Figure 11: EXAMPLE OF EACH COUNTER STAGE

## AMI\_TMRS (Counter Example)



## 5.2 INPUT & OUTPUT SIGNALS

- **CNTR\_DOUT<31:0>** Output Data Bus from this block.
- **LOCAL\_DIN<31:0>** Local gate array buffered version of the 68020 Data Bus used for write cycles.
- **LOCAL\_AIN<6:0>** Local gate array buffered version of the 68020 Address Bus. Used in local interface logic for low level address decoding.
- **CNT/TMR\_SEL\_L** Counter/Timer Logic select line.
- **LOCAL\_AS\_L** Local gate array buffered version of the 68020 Address Strobe signal.
- **LOCAL\_DS\_L** Local gate array buffered version of the 68020 Data Strobe signal.
- **LOCAL\_R/W\_L** Local gate array buffered version of the 68020 Read/Write direction control signal.
- **LOCAL\_34H\_CLK** 64nsec Clock that is derived from XCI\_34H.
- **LOCAL\_INIT\_L** Local gate array initialization.
- **WTCHDG\_INT\_H** Watchdog Timer Interrupt signal is asserted if software is unable to reset the counter.
- **RTOS\_TICK\_INT\_H** RTOS\_TICK is a hardware timer interrupt signal required by RTOS for its "TICK" timer routines.
- **BERR\_L** Bus Error signal that connects to 68020 to terminate a cycle in error.

### 5.2.1 WATCHDOG TIMER

The watchdog timer will employ a 32-bit down counter that provides a safeguard against software branching off into unknown space in a runaway code condition. The 32-bit counter running off the gate array's 64nsec clock will allow up a maximum delay of 274 seconds before issuing an interrupt. The length of time will be determined by the longest running uninterruptable task in the XFA application. The timer is loadable and has enable and clear inputs.

The Watchdog Timer will continually be reloaded before its programmed time period has a chance to expire. This is accomplished by 68020 software periodically writing to the WDTCLR Register. If the 68020 code fails to return and write the Register and reload the Watchdog Timer, it is assumed that a serious hardware failure has occurred and the timer will expire, generating an interrupt at IPL 7 to the 68020. Interrupt Level 7 is non-maskable such that the 68020 will always be able to receive this interrupt. The interrupt level 7 will cause the 68020 to vector to an Error Response Service Routine which will report the failure to the Host. The details of this routine are contained in the Software Specification. A pair of dedicated Error Status Line allows a parallel method of notifying the host of the error in the event that 68020 is unable to report the error.

### 5.2.1.1 WATCHDOG COUNTERS

The Watchdog Timer will be implemented with a 32-Bit down counter. The counter will be parallel loaded and will operate synchronously with the 64nsec master clock. A read from the same register address that is used to load the counters will provide current state of the counter.

An output from the counter will generate a state indicating that the programmed time has elapsed. This output will be directed to the interrupt circuitry as well as the Error Status logic.

### 5.2.2 BUS ERROR TIMER

The Bus Error Timer's purpose is to issue a BERRL signal should a 68020 Bus Cycle extend beyond a maximum specified time. This is to prevent a "hung" processor should code attempt to access non-existent memory or failed hardware does not return the proper DSACKL signals. The 68020 upon seeing the assertion of BERRL will terminate its current bus cycle and begin saving state to its stack. It will then begin Bus Error Exception Processing.

The way the Bus Error Counter works is as follows. Upon the start of every 68020 Bus Cycle, the BEC ( Bus Error Counter ) is enabled and allowed to count down from its programmed value. If the cycle terminates in the proper manner, the counter is reloaded and readied to start counting down at the beginning of the next valid bus cycle. If a 68020 Bus Cycle extends longer than the count down period, the BEC will expire and issue a BERRL control signal back to the 68020, terminating the bus cycle.

#### 5.2.2.1 BUS ERROR COUNTER

The Bus Error Counter is a 16-Bit Down Counter that runs off of the local 64nsec L34CLK. It can be programmed to a maximum expiration time of 4msec.

### 5.2.3 CONTROL CSR

Counter/Timer CSR ( CTCSR ) will contain control bits to enable and disable the Timers as well as put the counters into self test mode.

### 5.2.4 RTOS TIMER

The RTOS timer is required by the RTOS Operating System to establish a realtime clock timebase for software operations. It will periodically ( period TBD ) generate an interrupt to the 68020 at IPL 2. This Timer will be implemented with a 24-Bit Down Counter that is allowed to free run. The counters used to implement this function will operate off of the 64nsec local clock. The maximum period of time that is programmable is 1 second between interrupts generated.

# CHAPTER 6

## FDDI CHIPS CSR BUS INTERFACE

### 6.1 OVERVIEW

This interface contains the receiver control, driver control, and state machines necessary to interface the AMI Gate Array to the FDDI CSR Bus.

The state machine accesses the FDDI CSR Bus and transfer data between the addressed device on the bus and the 68020. A set of protocol control lines between this state machine and the the 68020 will provide a means of notification between processes. The state machine will be synchronized with the 80ns BYTCLK that the FDDI Chips operates from. The external data path is 16-Bits wide. Dynamic Bus Sizing will be generated here to reduce the bus to 16-Bits.

---

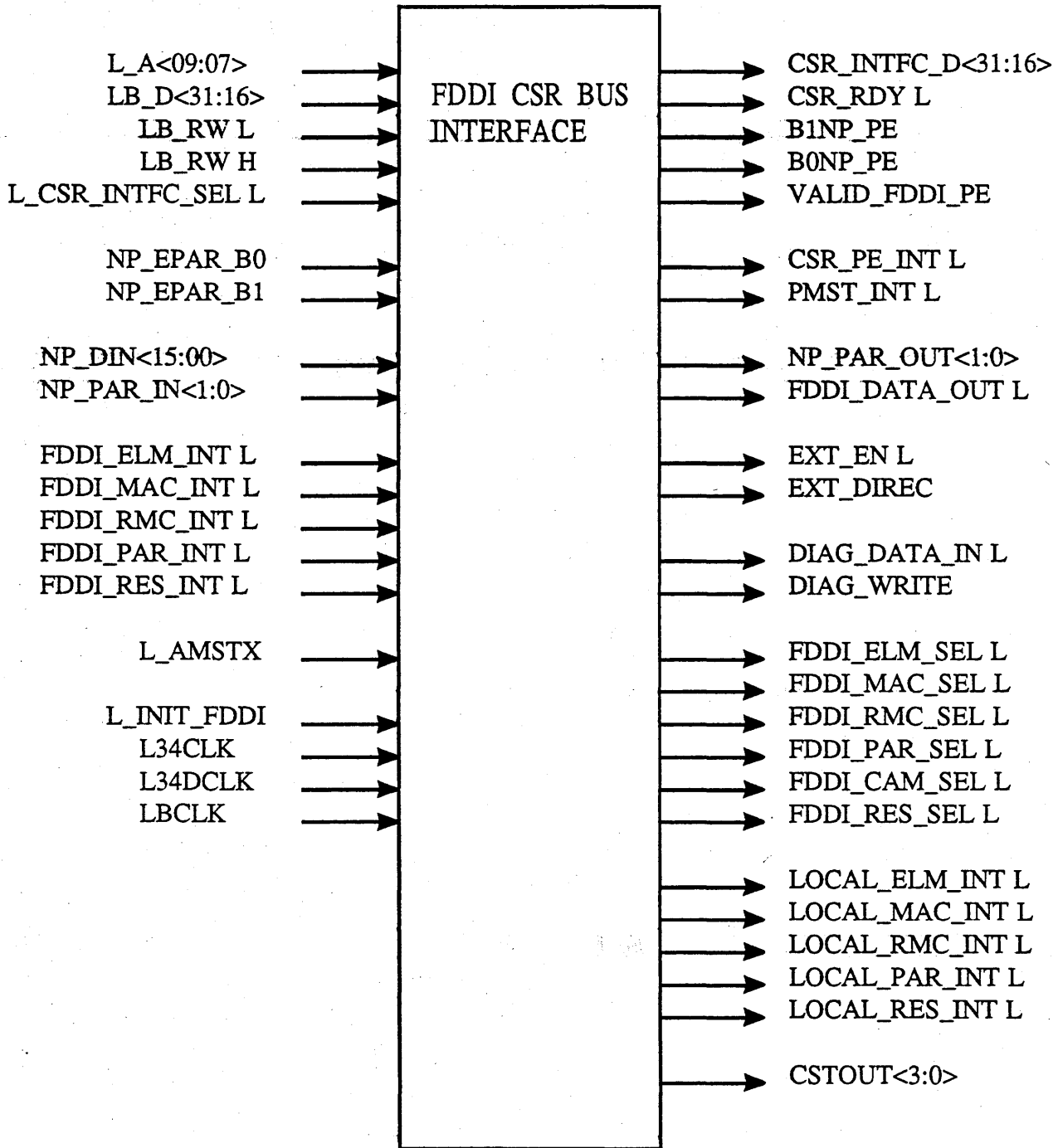
**Table 2: CSR TO BE ACCESSED**

---

| CSR                          | Address Space |
|------------------------------|---------------|
| RMC CSRs                     | 64 Words      |
| MAC CSRs                     | 64 Words      |
| ELM CSRs                     | 64 Words      |
| PARSER/CAM CSRs              | 64 Words      |
| MAC CAM CSRs                 | 64 Words      |
| RESERVED                     | 64 Words      |
| EXTERNAL DIAGNOSTIC REGISTER | 1 Word        |

---

Figure 12: CSR INTERFACE CONTEXT



### 6.1.1 CSR BUS DATA PATH & SIGNALS

- **NP\_DIN<15:00>**: This is the input data from the bidirectional three-state NP CSR DATA BUS used to exchange data between the FDDI chips and the AMI Gate Array.
- **NP\_PAR\_IN<1:0>**: This is the input buffered parity signal from the PARSER chip to be checked by the AMI gate array during a NP CSR BUS read cycle to the PARSER chip.
- **NP\_PAR\_OUT<1:0>**: This is the generated parity signal to be sent to the PARSER chip during a NP CSR BUS write cycle to the PARSER chip.
- **FDDI\_DATA\_OUT L**: This signal controls the data transceivers, on the AMI gate array, for the NP CSR BUS. A low on this signal will enable the AMI gate array to drive the NP CSR DATA BUS.
- **EXT\_EN L**: This signal is buffered and sent to the external data transceivers. When set low the external transceivers are enabled.
- **EXT\_DIREC**: This signal is buffered and sent to the external data transceivers. This signal is used for direction control.
- **FDDI\_ELM\_SEL L**: This output selects the ELM FDDI chip for the current NP CSR BUS cycle.
- **FDDI\_MAC\_SEL L**: This output selects the MAC FDDI chip for the current NP CSR BUS cycle.
- **FDDI\_RMC\_SEL L**: This output selects the RMC FDDI chip for the current NP CSR BUS cycle.
- **FDDI\_PAR\_SEL L**: This output selects the PARSER chip for the current NP CSR BUS cycle.
- **FDDI\_CAM\_SEL L**: This output selects the CAM chip for the current NP CSR BUS cycle.
- **FDDI\_RES\_SEL L**: This output selects the RESERVED chip for the current NP CSR BUS cycle. This signal is for future use.
- **FDDI\_ELM\_INT L**: This is the buffered input from the NP CSR BUS indicating an interrupt request from the ELM FDDI chip.
- **FDDI\_MAC\_INT L**: This is the buffered input from the NP CSR BUS indicating an interrupt request from the MAC FDDI chip.
- **FDDI\_RMC\_INT L**: This is the buffered input from the NP CSR BUS indicating an interrupt request from the RMC FDDI chip.
- **FDDI\_PAR\_INT L**: This is the buffered input from the NP CSR BUS indicating an interrupt request from the PARSER chip.
- **FDDI\_RES\_INT L**: This is the buffered input from the NP CSR BUS indicating an interrupt request from the RESERVED interrupt input.

### 6.1.2 CSR INTERFACE GATE ARRAY INTERNAL SIGNALS

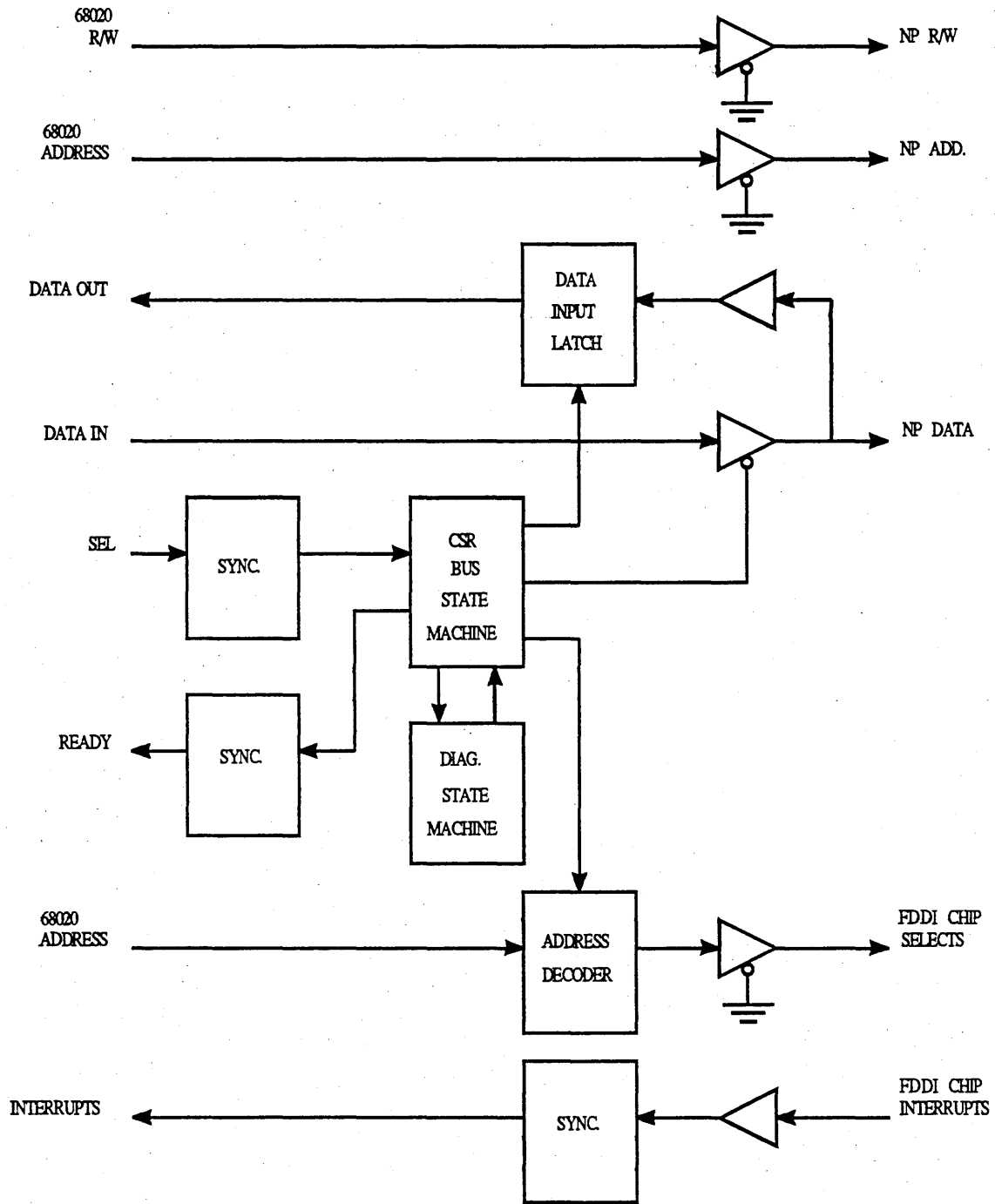
- **L\_CSR\_INTFC\_SEL L:** This signal tells the CSR INTERFACE that the current 68020 bus transaction is for this interface.
- **LB\_RW L:** This signal tells the CSR INTERFACE that the current 68020 bus transaction is a READ (LB\_RW L = 1) or WRITE (LB\_RW L = 0) cycle.
- **LB\_RW H:** This signal tells the CSR INTERFACE that the current 68020 bus transaction is a READ (LB\_RW = 0) or WRITE (LB\_RW = 1) cycle. This signal is the compliment of LB\_RW L.
- **L\_A<09:07>:** Local 68020 address bus will be decoded to tell the CSR INTERFACE which FDDI chip is being selected.
- **LB\_D<31:15>:** These signals contain the data to be written to the NP CSR BUS. It is used in this block to generate parity on a PARSER CSR write.
- **CSR\_RDY\_L:** This signal is translates to the Data Transfer and Size Acknowledge that is be supplied to the 68020. Due to the need for synchronization, this signal is be used to generate DSACKx.
- **CSR\_INTFC\_D<31:16>:** These signals are to be sent to the 68020 during a read cycle from the NP CSR BUS.
- **NP\_EPAR\_B0,NP\_EPAR\_B1:** These signals come from the GPCSR register (bits <21:22>). Setting these bits will force even parity checking on the NP CSR BUS during a PARSER read. This feature is supplied for SELF\_TEST use only. NP\_EPAR\_B0 is for byte 0 and NP\_EPAR\_B1 is for byte 1.
- **B0NP\_PE,B1NP\_PE:** These signals go to the GPCSR register (bits <19:20>). These signals flag which byte had a parity error on a NP CSR BUS read cycle. B0NP\_PE is for byte 0 and B1NP\_PE is for byte 1.
- **VALID\_FDDI\_PE:** This signal goes to the GPCSR register to signal when the byte parity error bits (B0NP\_PE,B1NP\_PE) are valid and can be latched by the GPCSR register.
- **CSR\_PE\_INT L:** This signal is sent to the interrupt control to flag a NP CSR BUS parity error interrupt.
- **LOCAL\_ELM\_INT L:** This signal is the synchronized version of FDDI\_ELM\_INT L that is sent to the interrupt control logic.
- **LOCAL\_MAC\_INT L:** This signal is the synchronized version of FDDI\_MAC\_INT L that is sent to the interrupt control logic.
- **LOCAL\_RMC\_INT L:** This signal is the synchronized version of FDDI\_RMC\_INT L that is sent to the interrupt control logic.
- **LOCAL\_PAR\_INT L:** This signal is the synchronized version of FDDI\_PAR\_INT L that is sent to the interrupt control logic.
- **LOCAL\_RES\_INT L:** This signal is the synchronized version of FDDI\_RES\_INT L that is sent to the interrupt control logic.
- **L\_AMSTX:** This signal is the buffered version of the Start Transmit interrupt sent to the AMI gate array by the PMC gate array. This signal is in the 80nsec time domain not the 64nsec time domain of the 68020.





### 6.3 CSR INTERFACE BLOCK DESCRIPTION

Figure 13: CSR INTERFACE - BLOCK DIAGRAM



### **6.3.1 SYNCHRONIZERS**

These blocks are used to synchronize L\_CSR\_INTFC\_SEL L, CSR\_RDY, and interrupt requests between the CSR BUS (80ns time domain) and the LOCAL Gate Array bus (64ns time domain). An extremely conservative dual rank synchronizer is used.

### **6.3.2 ADDRESS DECODER**

This block is used to decode the address lines to determine which chip is being selected. After decode the proper chip is be selected for access.

### **6.3.3 CSR BUS STATE MACHINE**

This block contains the state machine that will control the arbitration used to do a read/write to the CSR Bus for chip accesses. This block will also control the interfaces connection to the rest of the AMI internal blocks.

### **6.3.4 DIAGNOSTIC STATE MACHINE**

This block contains the state machine that will control the arbitration used to do a read/write to the CSR Bus for external diagnostic register accesses.

### **6.3.5 DATA INPUT LATCH**

This block contains latches used for the CSR bus data on a read operation from the CSR bus. This data is supplied to the 68020.

# CHAPTER 7

## AMI/ESP INTERFACE

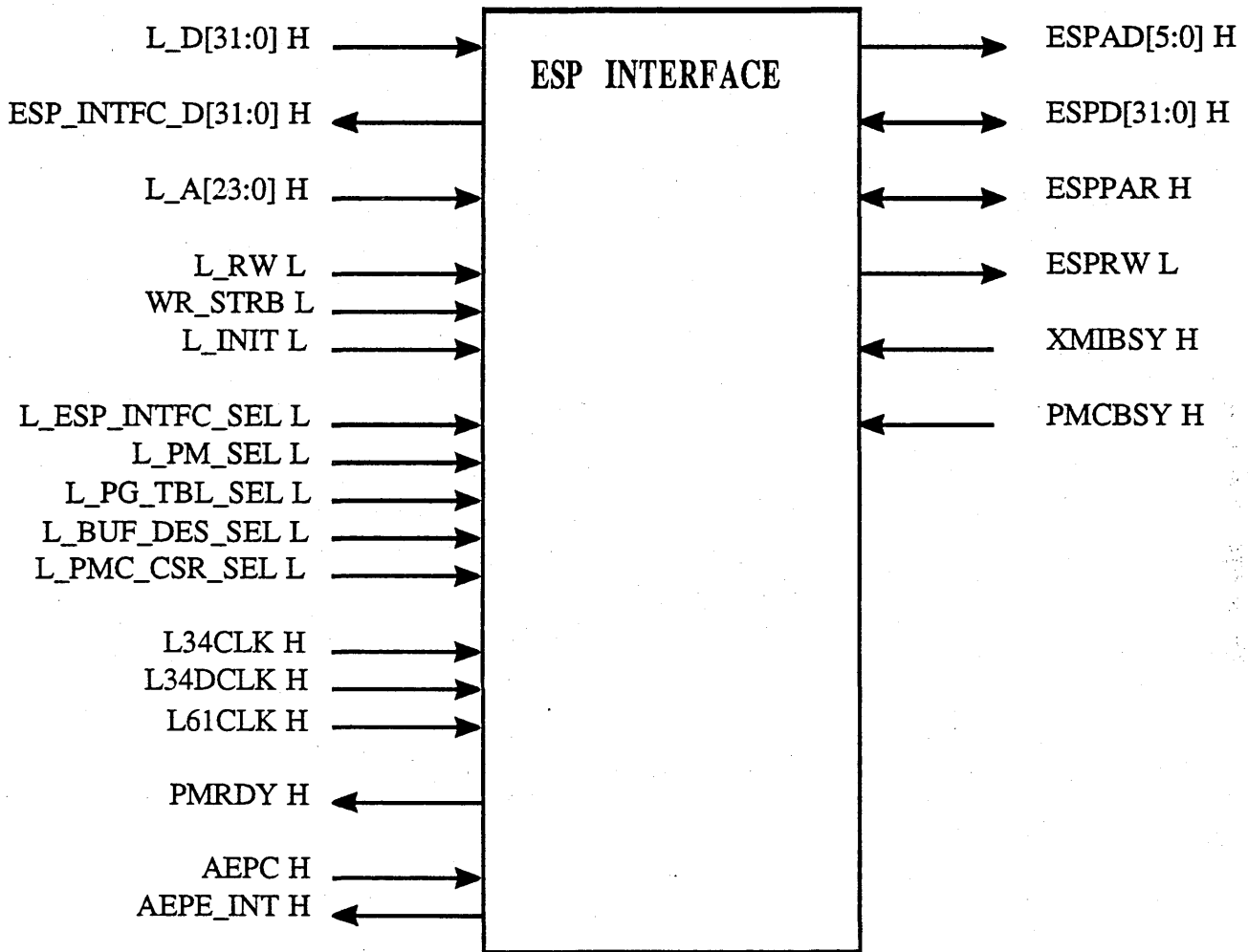
### 7.1 OVERVIEW

The AMI/ESP Interface is a channel that provides communication between the AMI Subsystem and the ESP Gate Array. The data channel is 32-Bits wide. All transfers will be longword operations. The ESP Gate Array internally provides a communication path to the Host as well as the Packet Memory Controller Gate Array.

Within the ESP are registers that are visible by the AMI Interface. These registers are the means by which the AMI communicates with the Host and the PMC. Communication is indirect in that one register holds the Command/Address and other registers hold the Data to be transferred. There is a long list of other registers within the ESP that are accessible by the AMI. The details of these registers can be found in the ESP Design Specification.

Two modes of operation will be provided for in the AMI/ESP Interface. The first mode can be viewed as a Simple Programmed I/O access of the registers within the ESP Chip. In this mode the 68020 can read and write registers within the ESP Chip. This mode is mainly for use in communicating with the ESP/XMI Registers. The second mode of operation allows the 68020 to directly address the Packet Memory, Page Table Entries, Buffer Descriptors and PMC CSRs. This mode will be accomplished by having a state machine perform hidden AMI/ESP Bus cycles while holding off the assertion of the 68020's DSACKL[1:0] signal lines. The 68020's Bus Cycle will be held off until the AMI/ESP Bus Cycles are completed. In this manner, the PM Subsystem will appear directly mapped into 68020 Address Space.

Figure 14: AMI/ESP INTERFACE CONTEXT



### 7.1.1 AMI/ESP INTERCONNECT BUS

- **ESPAD<5:0> H:** These 6 address lines, between the AMI and ESP, provide the ability to address up to 64 longword locations within the ESP Gate Array.
- **ESPD<31:0> H:** This is the data bus connecting the AMI and the ESP. It is a bi-directional tri-stateable data bus.
- **ESPPAR H:** Longword Parity is generated and checked on the ESP Bus via this parity bit. Odd bus parity protection will be supported.
- **ESPRW L:** The Read/Write Line defines the direction of the data transfer.
- **XMIBSY H:** This signal asserted indicates to the AMI that the last transaction that was requested thru the AMI/ESP Interface intended for the XMI Host has not completed yet and is still in-process. The next transaction intended for the XMI should not start until the BUSY Bit is de-asserted.
- **PMCBSY H:** This signal asserted indicates to the AMI that the last transaction that was requested thru the AMI/ESP Interface intended for the PMC has not completed yet and is still in-process. The next transaction intended for the PMC should not start until the BUSY Bit is de-asserted.

### 7.1.2 AMI INTERNAL SIGNALS

- **L\_D<31:0> H:** Local AMI Data Bus that is received by the AMI from the 68020.
- **ESP\_INTFC\_D H<31:0>:** Local AMI Data Bus that is driven by AMI/ESP Interface.
- **L\_A<23:0> H:** Local AMI Address Bus that is received from the 68020.
- **L\_ESP\_INTFC\_SEL L:** AMI/ESP Interface Chip Select is a signal decoded off the Address Bus and is qualified with 68020 ASL. It comes from the common G.A. Address Decode Logic and memory maps the AMI/ESP Register Interface.
- **L\_PM\_SEL L:** AMI/ESP Interface Chip Select is a signal decoded off the Address Bus and is qualified with 68020 ASL. It comes from the common G.A. Address Decode Logic and memory maps the directly addressable Packet Memory Space.
- **L\_PG\_TBL\_SEL L:** AMI/ESP Interface Chip Select is a signal decoded off the Address Bus and is qualified with 68020 ASL. It comes from the common G.A. Address Decode Logic and memory maps the directly addressable PM Page Table Space.
- **L\_BUF\_DES\_SEL L:** AMI/ESP Interface Chip Select is a signal decoded off the Address Bus and is qualified with 68020 ASL. It comes from the common G.A. Address Decode Logic and memory maps the directly addressable PM Buffer Descriptor Space.
- **L\_PMC\_CSR\_SEL L:** AMI/ESP Interface Chip Select is a signal decoded off the Address Bus and is qualified with 68020 ASL. It comes from the common G.A. Address Decode Logic and memory maps the directly addressable PMC's CSR Space.
- **L\_RW L:** Local Read or Write bus cycle direction control line.
- **WR\_STRB L:** Write Strobe signal is used to validate data on the local data bus. This signal is generated from the AMI Control Logic.
- **L\_INIT L:** Local Initialization signal is asserted during a reset condition to set all logic to a known reset state.

- **AEPE\_INT H:** AMI/ESP Bus Parity Error Interrupt is an output signal from the AMI/ESP Bus Interface Parity Generator/Check Logic. This interrupt will be issued on a parity mismatch on an ESP Bus Read Cycle. Parity Generation and Checking for the AMI/ESP Bus defaults to ODD Parity.
- **AEPC H:** AMI/ESP Bus Even Parity Check Bit is a control bit from the General Purpose CSR (GPCSR). This bit when set to a logic "1" will enable EVEN Parity checking on AMI\ESP Bus reads. The default value of this bit is at logic "0" and selects ODD Parity Checking. The default parity on the AMI/ESP Bus is ODD Parity. This control bit is primarily for testing the bus parity circuitry.
- **L34CLK H:** This is a buffered version of a 64nsec clock that is supplied by the XMI Corner.
- **L34DCLK H:** This is a buffered version of delayed version of L\_34CLK H. The delay is 10.7nsec.
- **L61CLK H:** This is a buffered version of a 64nsec clock that is supplied by the XMI Corner.
- **PMRDY L:** This is the local AMI/ESP Bus Interface version of DSACKL. The signal is used to notify the 68020 that the AMI/ESP Interface has acknowledged the data and that the current cycle should be completed.

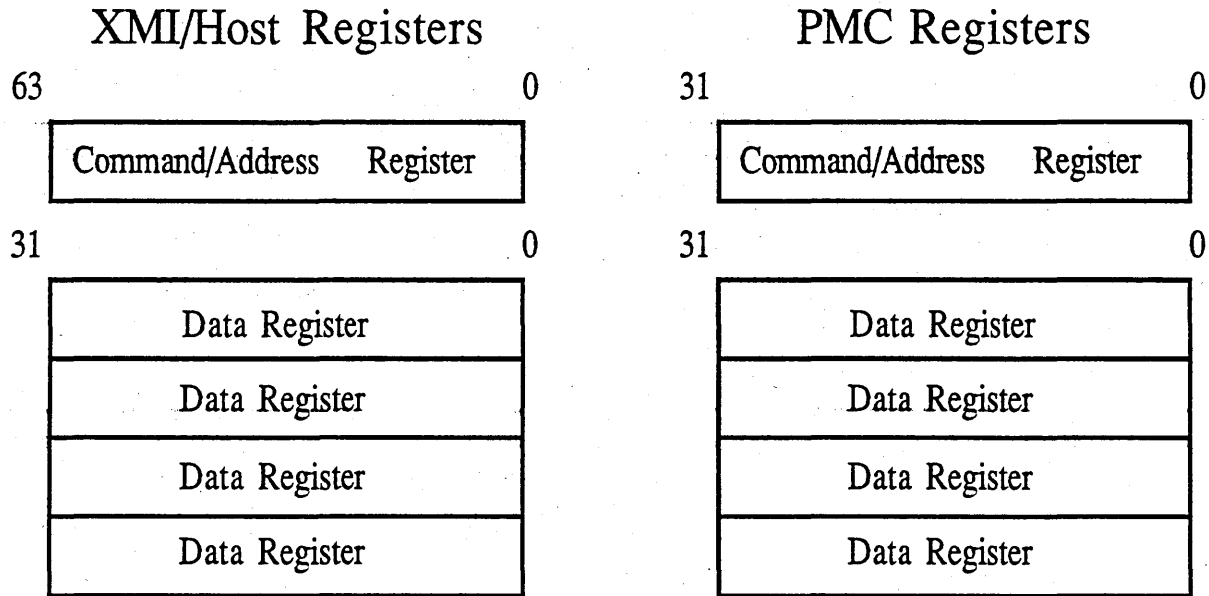
### 7.1.3 AMI/ESP INTERFACE FUNCTIONS

The AMI/ESP Interface Block will allow simple short transfers from the 68020 to the ESP through an interface that will tightly couple the 68020's bus cycle to the ESP Bus cycle. The 68020 will have the ability through this interface to perform read and write cycles to the ESP registers. A group of addresses will be provided to map the ESP Registers into 68020 space. The 68020 bus protocol will be modified through this interface to provide the ESP with the bus protocol that it expects. The 68020 will know when a ESP Transaction has completed by polling the appropriate Busy Bit in the ESP Interface Status Register (ESPIFSR).

Other logic will provide the ability to directly access Packet Memory, PM Page Table, PM Buffer Descriptors and PMC CSRs. These Packet Memory sections will be memory mapped into the 68020's memory space. The act of accessing one of these addresses will start a state machine that will perform accesses over the AMI/ESP Bus to transfer data.

**Figure 15: AMI/ESP Transaction Registers**

Registers contained within ESP Chip



**7.1.4 ADDRESS DECODER and CONTROL**

This block contains the logic to decode local 68020 address, select lines, reset signals, control signals and produces the control signals needed for the various registers and logic within the AMI/ESP Interface. This block works together with the AMI/ESP Bus Access State Machine to determine when a simple register read/write is enabled vrs a PM direct addressed sequence.

**7.1.5 PMC COMMAND/ADDRESS FORMATTER**

This section of logic merges the local 68020 address into a Command/Address field for the PMC. The AMI Cycle Control selects one of four areas: Packet Memory Space, Page Table Entries, Buffer Descriptor Space and PMC CSR Space. The local 68020 address is translated into a PMC C/A format through hard wired muxes and is setup to be written to the Adapter Manager Packet Buffer Address Register ( PMCRBAR ) in the ESP Chip. This would occur through an AMI/ESP Bus Interface C/A Write cycle.



### 7.1.6 PM READ - ADDRESS CACHE

When the 68020 performs a read from the PM Address Space, logic compares the address with a latched version of the last octaword address obtained in the last read access of PM. The latched octaword address is a form of address caching to minimize the number of PMC Transactions on contiguous addressing. If a match occurs, then the AMI/ESP Bus Access State Machine will obtain the data from one of the ESP's Adapter Manager Packet Buffer Data Register. This data will be bused out to the 68020 and the state machine will complete the cycle by asserting PMRDY L.

If no match occurs, the AMI/ESP Bus Access State Machine will write the PM translated Command/Address to the ESP and wait for PMCBSY H to de-assert. Upon the completion of the PMC Transaction, the ESP will deassert PMCBSY H allowing the AMI/ESP Bus Access State Machine to access the AMPBD0R thru AMPBD3R Registers and obtain the data to be applied to the 68020's data bus. The cycle is completed by the AMI/ESP Bus Access State Machine asserting PMRDY L.

### 7.1.7 AMI/ESP CYCLE CONTROL

This control logic is responsible for the directly addressed accesses of: Packet Memory Space, Page Table Entries, Buffer Descriptor Space and PMC CSR Space. It also will determine if a local address cache hit or miss occurs on a PM Read Cycle. It will be responsible for invalidating the cache if a register access of Adapter Manager Buffer Address Register ( AMPBAR ) occurs in the other interface modes.

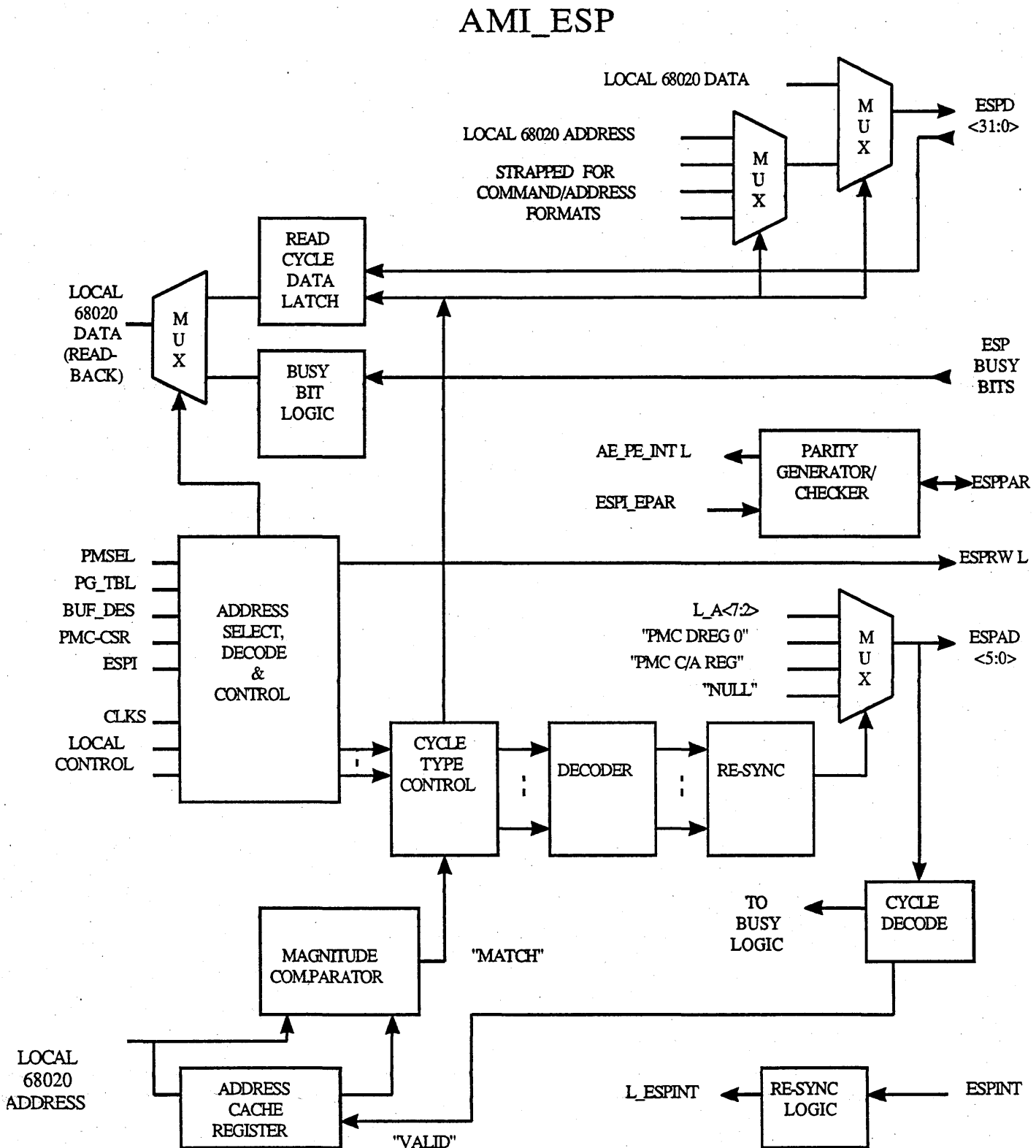
### 7.1.8 AMI/ESP STATUS REGISTER - ESPIFSR

This register provides the status of the ESP interface busy bits for transactions to both the XMI and PMC. The two bits represent the state of transactions that are in process to the XMI and PMC. This register must be read by software and until the status is read to be cleared, no further access of the associated port should be taken by software. These status bits are only meaningful in the simple register access mode.

### 7.1.9 AMI/ESP INTERFACE PARITY

Longword parity will be generated on a write to the ESP Bus and Longword parity will be checked against the data on the ESP Bus during a longword read cycle. Parity will be checked and generated in logic local to the AMI/ESP Interface. In the instance of a parity error occurring, a flag is set and generates a state on the AMI/ESP Error Reporting Lines indicating a Broken Bus. An ESP Interface Parity Error interrupt ( IPL 7 ) is generated to the 68020 at the same time the error codes are generated. This error is Fatal. Odd parity is the default for the AMI/ESP Bus, but provisions are made through the GPCSR Register to set Even Parity Checking to allow testing of the logic.

Figure 16: ESP INTERFACE - BLOCK DIAGRAM



### 7.1.10 LATCHES

Transparent data latches are used as temporary storage in the data read path to assist in retiming transfers from AMI/ESP Bus to 68020. Data from the 68020's data bus is driven to the AMI/ESP Bus as the default condition. This was based on early information that tri-stated bidirectional Gate Array buses caused problems, thus this bus is driven at all times except when a read cycle is in process.

### 7.1.11 MUXES

Muxes are used in the address and data path to the ESP to allow either a simple direct access by the 68020 or hidden cycles in a direct addressed mode access of the PM Subsystem. In the Address Path to the ESP Bus another input to the mux is a Null Address ( all zeros ). This is placed on the AMI/ESP Bus as a default address when the ESP Bus is not in use!

Muxes in the data path to the 68020 are for an ESP READ transaction or reading the status of the busy bits in the ESPIFSR register.

### 7.1.12 PAGE OWNERSHIP

Page Ownership will be contained in the Most Significant Bit<31> of the Page Table Entry Data Field on a read. Also contained in the PTE Data Field will be the Page Number, Two Color Bits and the Forwarding Vector. Bit<31> can be used to determine page ownership. Another method of determining page ownership is to read the ESP ERROR REGISTER and test Bit<9>. This bit will be set if the 68020 owns the current packet buffer page that it is accessing, and cleared if the 68020 does not own the accessed page.

Software is responsible for determining buffer ownership.



### 7.1.13 CLEAR OWNERSHIP - DIRECT ADDRESSED PACKET MEMORY ONLY!

The default is to clear ownership of a Buffer after a Buffer Descriptor Write. This is accomplished through the direct mapped operation only. This automatic clear ownership after a Buffer Descriptor Write requires a software order to control when the first Buffer Descriptor gets written. Software writes the first Buffer and then the second Buffer and then the second Buffer Descriptor and then the third Buffer followed by the third Buffer Descriptor and so on and so forth until all buffers and descriptors are written. The final operation would be to write the first Buffer's Buffer Descriptor thus allowing the following stage to complete a transaction. Until the first Buffer is seen to have changed ownership, no action is taken upon it and the following buffers.

The default setting can be changed through a CSR bit for diagnostic purposes. The other clear ownership cycle will require a separate write to the ESP's PMC Command/Address Register with a properly formatted data word. No other direct mapped clear ownership operation is provided.

#### 7.1.14 Buffer Descriptor, Page Table Entry and PMC CSR

These read/write operations are single longword only!

Accesses to the Page Table Entries are on even 512 Byte boundaries as the lower 68020 Address bits A<8:0> are truncated and not used. In general for a PTE access 68020 Address Bits A<8:0> must be zero to avoid confusion.

# CHAPTER 8

## INTERRUPT CONTROLLER

### 8.1 OVERVIEW

The 68020 microprocessor can support seven hardware levels of interrupt. The internal priority encoder assigns the highest priority interrupt to level seven (IPL7) and the lowest to level one (IPL1); level zero corresponds to no interrupt requested. Interrupts are prioritized in the AMI Gate Array. Multiple interrupt sources may occupy the same priority level. This Sublevel Priority Encoding occurs prior to 68020 IPL priority decode.

All IPL levels and interrupt sources below level seven can be Enabled, Disabled and Masked by the 68020. These functions are implemented for debug and Self Test. IPL seven interrupts can not be blocked. This level of interrupts is reserved for events requiring immediate attention.

Each source of interrupt provides a unique vector to the 68020 on an Interrupt Acknowledge Cycle. The vector registers are programmable by the 68020. Two types of exception processing are supported by the MC68020. They are, Interrupt Recognition and Bus Error.

#### 8.1.1 Interrupt Sources

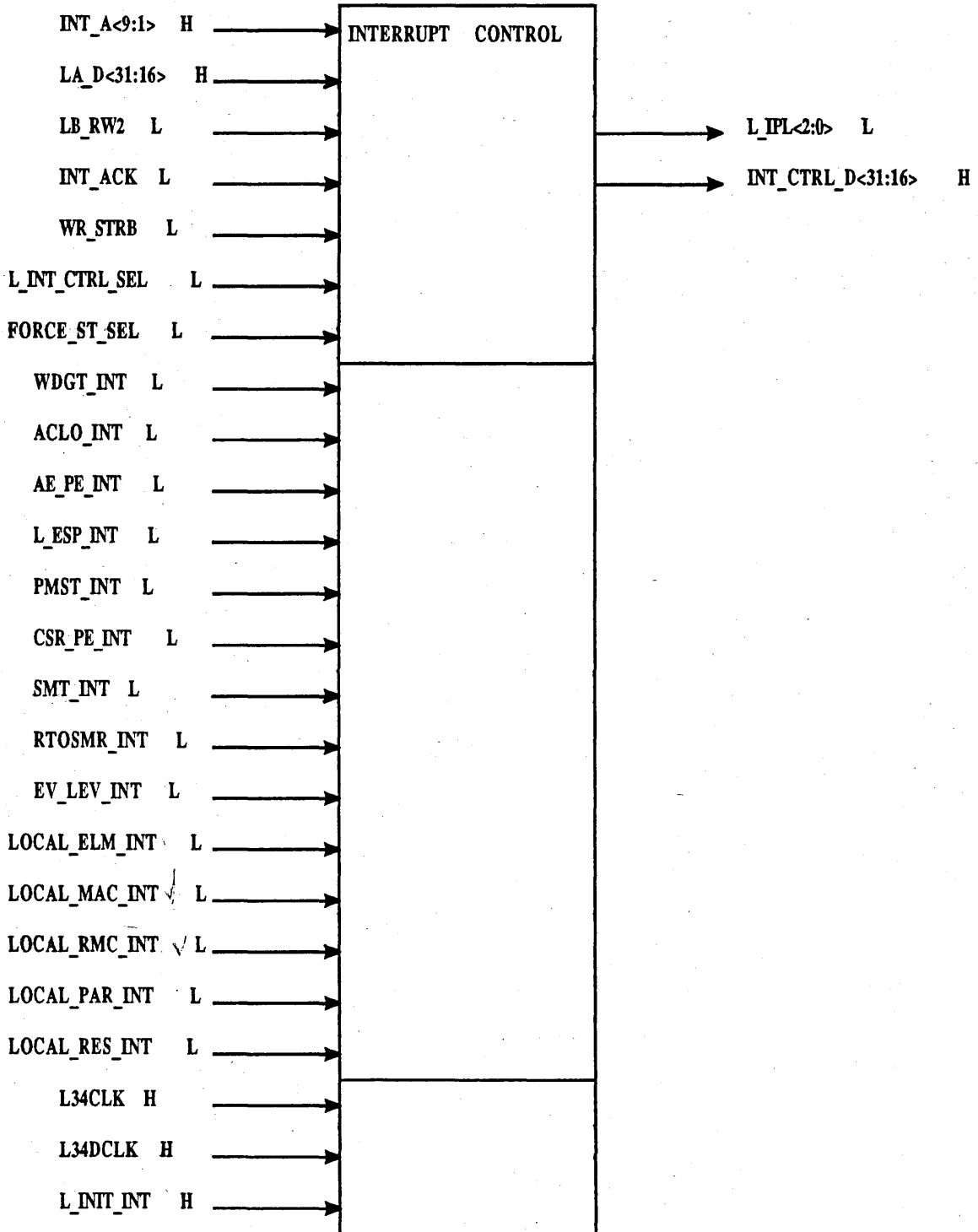
- **FDDI Chips**

- RMC Chip*
  - Reserved*
  - MAC Chip*
  - ELM Chip*
  - PARSER/CAM*
  - FDDI CSR Bus Parity Error*

- **SMT Priority** 2
- **RTOS Timer** 3
- **Watchdog Timer** 4
- **AC Power Fail** 5
- **Event Level Monitor** 6
- **ESP/Host** IPL 7
  - ESP*
  - PMC Start Transmit*

*ESP Bus Parity Error*

**Figure 17: INTERRUPT CONTROL CONTEXT**



### 8.1.2 INTERRUPT SOURCE SIGNALS

- **WATCHDOG\_INT\_L:** This signal indicates that the Watchdog Timer is requesting an interrupt.
- **ACLO\_INT\_H:** This signal is an interrupt indicating that an AC Power Fail is occurring.
- **AE\_PE\_INT\_L:** This signal indicates that a parity error has occurred on a ESP chip transaction and is requesting an interrupt to notify the processor.
- **L\_ESP\_INT\_L:** This signal indicates that the ESP interface is requesting an interrupt to service an ESP chip request.
- **PMST\_INT\_L:** This signal is an interrupt indicating PMC has a packet for Adapter Manager.
- **CSR\_PE\_INT\_L:** This signal indicates that a parity error has occurred on a PARSE chip transaction on the FDDI CSR BUS, and is requesting an interrupt to notify the processor.
- **SMT\_INT\_L:** This signal indicates an SMT priority interrupt.
- **RTOSMR\_INT\_L:** This signal indicates that a RTOS timer is requesting an interrupt.
- **EV\_LEV\_INT\_L:** This signal indicates that the Event Level Monitor is requesting an interrupt.
- **LOCAL\_ELM\_INT\_L:** This signal indicates that the ELM chip is requesting an interrupt.
- **LOCAL\_MAC\_INT\_L:** This signal indicates that the MAC chip is requesting an interrupt.
- **LOCAL\_RMC\_INT\_L:** This signal indicates that the RMC chip is requesting an interrupt.
- **LOCAL\_PAR\_INT\_L:** This signal indicates that the PARSE/CAM chip is requesting an interrupt.
- **LOCAL\_RES\_INT\_L:** This signal is a reserved interrupt for the FDDI CSR BUS.

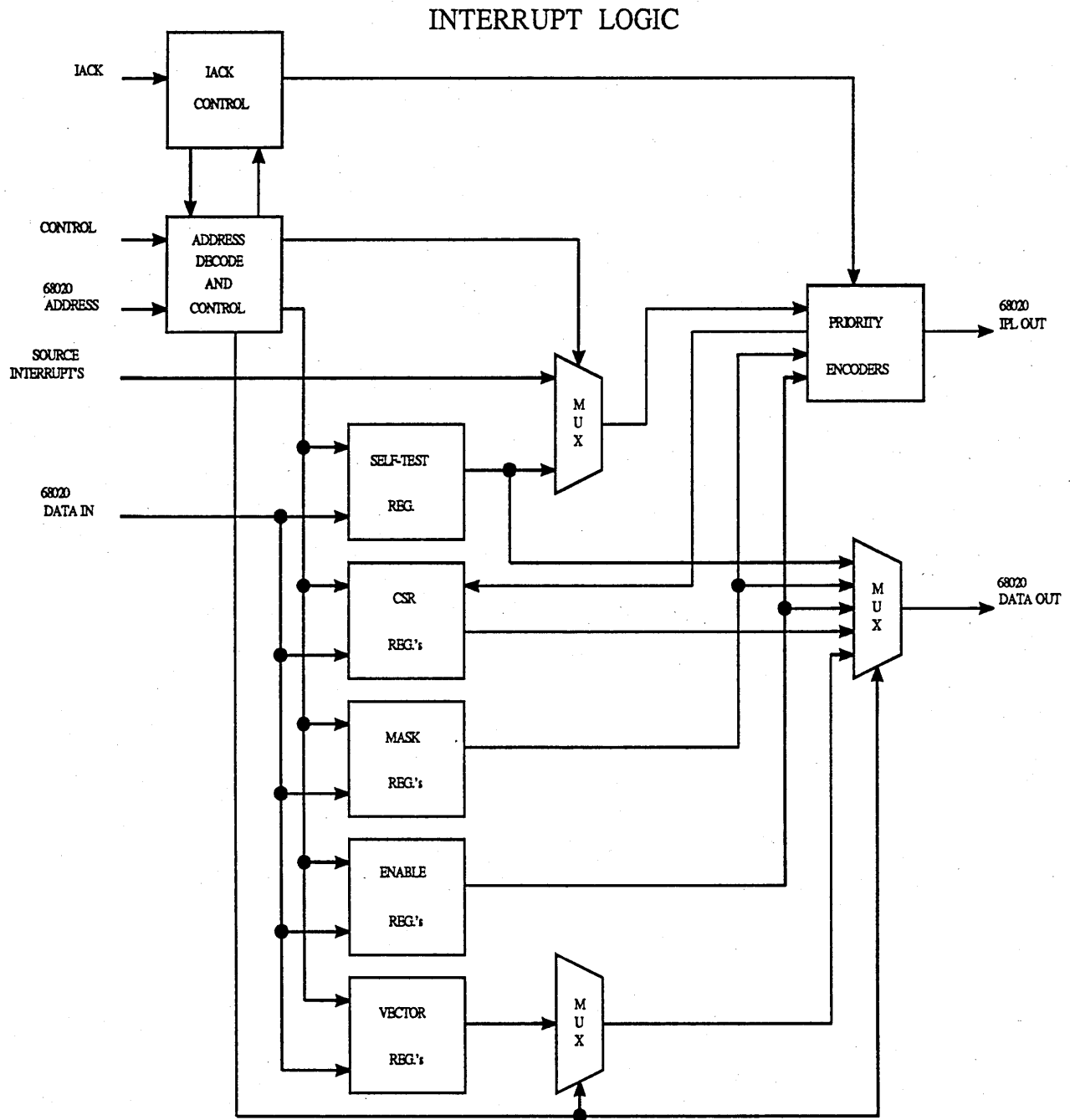
### 8.1.3 INTERRUPT CONTROLLING SIGNALS

- **INT\_A<09:01>:** These are internal address lines used to select Interrupt Controller registers and verify priority levels during an IACK cycle.
- **LA\_D<31:16>:** These are internal input data lines used to load Interrupt Controller register information.
- **INT\_CTRL\_D<31:16>:** These are internal output data lines used to supply vector and Interrupt Controller register data to the 68020.
- **LB\_RW2\_L** This signal tells the Interrupt Control block that the current 68020 bus transaction is a READ (LOCAL\_R/W = 1) or WRITE (LOCAL\_R/W = 1) CYCLE.
- **INT\_IACK\_L** This signal tells the Interrupt Control block that an Interrupt Acknowledge cycle has started. The proper IACK will be generated to clear the source of the interrupt.
- **WR\_STRB\_L** This signal tells the Interrupt Control block that data on the LOCAL\_DIN lines are valid.

- **L\_INT\_CTRL\_SEL\_L** This signal tells the Interrupt Control block that the current 68020 READ/WRITE bus cycle is for this block.
- **FORCE\_ST\_SEL\_L** This signal is used to select the INTERRUPT SELF TEST REGISTER as the source of interrupts.
- **L\_IPL<2:0>\_L** This signals are the interrupt priority levels (IPL) lines sent to the 68020.
- **L\_INIT\_INT** This signal is the Local gate array initialization flag.
- **L34CLK** This signal is the local buffered version of XMI Clock XMI 34H, and has a 64ns period.
- **L34DCLK** This signal is the local buffered version of 10.7nsec delayed XMI Clock XMI 34H, and has a 64ns period.



Figure 18: INTERRUPT CONTROL DETAIL BLOCK



### 8.1.4 Priority Encoder Logic

This is the logic that provides a priority encoded output based on all the various interrupt inputs. A priority level is assigned to groups of inputs since there are more interrupt sources than there are Interrupt Priority Levels, (IPL's) on the 68020. Logic will prevent the outputs from changing while IACK is asserted. A description of the nature and handling of these interrupts will be discussed later in this chapter.

The following list shows the source interrupts and the corresponding IPL's.

**Table 4: INTERRUPT STRUCTURE**

| IPL | DESCRIPTION        | PRIORITIZED SOURCE                                                               |
|-----|--------------------|----------------------------------------------------------------------------------|
| 7   | System Error, RTOS | Watchdog Timer<br>AC Power Fail<br>AMI/ESP Bus Parity Error (AMI Read operation) |
| 6   | XFA                | ESP Chip<br>FDDI CSR BUS PARITY ERROR                                            |
| 5   | FDDI Interface     | ELM<br>PARSER                                                                    |
| 4   | FDDI Interface     | MAC<br>RES<br>RMC<br>SMT Priority                                                |
| 3   | XFA                | PM Start Transmit                                                                |
| 2   | System Required    | Hardware Timer                                                                   |
| 1   | RTOS Required      | Event Level Monitor                                                              |

This block also sends information to the Address Decode and Control block. This information is used to select the proper vector for the interrupt. These vectors are contained in the Vector Registers.

This logic will also control the Mask and Enable/Disable function to source and IPL interrupts.

### 8.1.5 Address Decode and Control

This block will select the proper vector in the Vector Registers to be sent to the 68020. While IACK is asserted the output address to the Vector Registers will not change. This will assure that the correct vector is issued for the interrupt to be serviced. Since various interrupts can occur at the same IPL level, the vector is used to distinguish between interrupt sources.

This block is also used to decode the address input and, when the Interrupt Control is selected, will select and control the appropriate register for a 68020 read or write operation.

### 8.1.6 Interrupt Acknowledge Logic

This logic decodes the interrupt acknowledge cycle of the 68020 and directs the, Address Decode and Control block, and the Priority Encode block during an Interrupt Acknowledge cycle. This block helps in the formulation of the vector register address, and determines which source and IPL pending bit is cleared in CSR1 and CSR2.

### 8.1.7 Self-Test Register

This register is supplied for self-test purposes. All source type interrupts can be emulated through this register. This register will operate only when the Self-Test Select bit (bit<14>) of the AMI General Purpose CSR is set.

### 8.1.8 CSR Registers

This logic contains interrupt status registers.

There is an interrupt pending status register for source interrupts (CSR1) and IPLs (CSR2) to identify which interrupts are pending. As an interrupt is serviced by an IACK cycle the corresponding bit in this register will be cleared. If the interrupt is masked this register will contain which interrupt source occurred while being masked (CSR1) and which IPL level this masked interrupt resides in (CSR2). This operation allows the use of polled mode software. Writing a one to the corresponding bit in the source interrupts pending register (CSR1) will clear the pending bit and its associated IPL pending register (CSR2); provided no other source is pending at the same IPL level.

### 8.1.9 Interrupt Mask Register

This register is used to mask in hardware individual interrupt channels. An interrupt can come in and be latched, but be prevented from propagating the request through to the 68020 as long as the mask bit for the channel is asserted. By selectively masking interrupts, the channels are in-effect re-prioritized.

Two registers are supplied to accomplish the masking operation, Source Interrupt Mask Register (SIMR), and IPL Interrupt Mask Register (IPLIMR).

### 8.1.10 Interrupt Enable/Disable Register

These registers will allow a programmer to enable or disable individual interrupt sources (SIER register) or IPL levels (SIPLIER register). A disabled interrupt channel is completely inactive - thus interrupts received on the channel are ignored and never seen by the 68020.

All IPL levels and interrupt sources below level seven can be Enabled, Disabled and Masked by the 68020. IPL seven interrupts can not be blocked. This level of interrupts (IPL seven) are reserved for events requiring immediate attention.

### 8.1.11 Vector Registers

This is a bank of programmable registers that hold the vectors that will be returned to the 68020 for each different interrupt acknowledge. The 68020 code has the flexibility to store any vector desired associated with any interrupt level. Vectors are Byte ( 8-Bits ) wide and are read on D<31:24>. The port is word wide and bits D<23:16> are read as zero and ignored during a read operation. The 68020 expects to read a vector from 64 to 255.

Upon receipt of the Vector, the 68020 will multiply this vector by 4 to form the vector offset, which is added to the vector base register to obtain the address of the vector in a longword.

## 8.2 Nature and Handling of Interrupts

There are two types of interrupts handled by the Interrupt control logic, Level sensitive and Edge triggered interrupts.

Level sensitive interrupts are caused by the RMC, MAC, ELM, Parser, RES, and ESP source interrupts. These interrupts will remain asserted until the source of the interrupt is cleared during the interrupt service routine.

Edge triggered interrupts are pulses sent to the Interrupt control logic by SMT, RTOS, Watchdog, AC Power Fail, Event Level Monitor, PM Start Transmit, FDDI CSR bus parity error, and ESP bus parity error source interrupts. These interrupts will be cleared during an IACK cycle that pertains to the individual source interrupt being acknowledged.

## 8.3 Interrupt Latency

Table 5: Worst Case Interrupt Latency

| CAUSE               | CLOCK CYCLES | COMMENT                                                                                                                                                                 |
|---------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 68020               | 48 (2/2/8)   | Interrupt, M-STACK.<br>2 read cycles, 3 clocks per read.<br>2 access cycles, 3 clocks per access.<br>8 write cycles, 3 clocks per write.<br>12 MC68020 internal clocks. |
| CURRENT INSTRUCTION | 91           | This is due to the fact that the 68020 will complete the current instruction before servicing the interrupt. This number is for the worst case DIVS.L instruction.      |
| SYNCHRONIZATION     | 2.5          | CSR chip interrupt.                                                                                                                                                     |
| <b>TOTAL</b>        | <b>141.5</b> | <b>141.5clks * 64nsec = 9.056usec</b>                                                                                                                                   |

**Table 6: Typical Interrupt Latency**

| CAUSE               | CLOCK CYCLES | COMMENT                                                                                                                                                                 |
|---------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 68020               | 48 (2/2/8)   | Interrupt, M-STACK.<br>2 read cycles, 3 clocks per read.<br>2 access cycles, 3 clocks per access.<br>8 write cycles, 3 clocks per write.<br>12 MC68020 internal clocks. |
| CURRENT INSTRUCTION | 8            | This is due to the fact that the 68020 will complete the current instruction before servicing the interrupt. This number is for the typical instruction time.           |
| SYNCHRONIZATION     | 2.5          | CSR chip interrupt.                                                                                                                                                     |
| <b>TOTAL</b>        | <b>58.5</b>  | <b>58.5clks * 64nsec = 3.744usec</b>                                                                                                                                    |

# CHAPTER 9

## RESET LOGIC

### 9.1 OVERVIEW

This RESET logic is required to bring the 68020 Subsystem to a know start-up state. It insures that all logic circuits in the subsystem have been initialized to the reset state prior to program code execution.

At power up a reset signal is generated to the MC68020 that holds its RESET pin low for a minimum of 100msec after +5V and Clock are stable. The 68020 will read from address \$00000000 after RESET\_L changes state from asserted to deasserted. At address \$00000000 the 68020 will fetch its Stack Pointer and at \$00000004 it will fetch its Program Counter. Program execution then begins at the address loaded into the Program Counter.

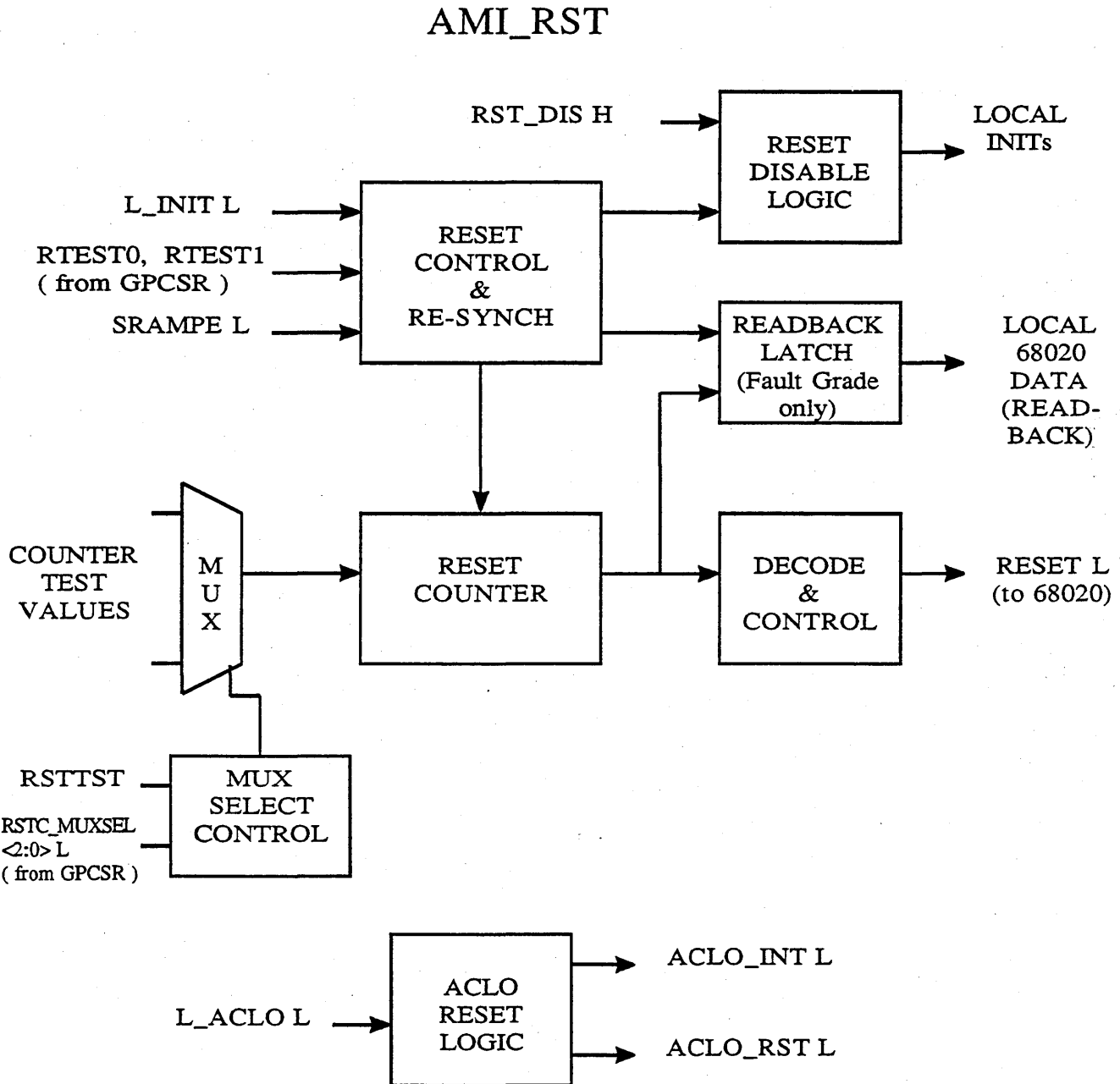
Since the XMI Spec only guarantees the XMI Clock to be stable for 1msec prior to the deassertion of DCLO, a lengthened reset signal must be generated to provide the 101msec long 68020 RESET signal. This delay requires a 21-Bit counter be used to count the 64nsec clock cycles and generate a 101msec delay for use in generating 68020 RESET signal.

The Reset Instruction from the 68020 is not supported by the AMI logic. The AMI always drives the RESET L signal line.

SRAM Parity Error will cause a 68020 Reset and hold it in that state until a Host generated init occurs by pulsing the INITL line.

A Reset Disable Bit<13> is provided in the GPCSR to allow the 68020 to block an INITL assertion and deassertion from resetting the AMI and the 68020 while resetting everything else on the Adapter. The Reset Disable Bit gets disabled after the INITL pulse has occurred.

Figure 19: Reset Logic Block Diagram



# CHAPTER 10

## MEMORY SUBSYSTEM

### 10.1 BUFFERS & TRANSCEIVERS

The 68020 Address Bus and Data Bus are buffered before driving the memory subsystem. This is done to provide added current drive capability required when driving the memory array capacitance. This also allows the option to terminate the memory buses in the etch characteristic impedance to minimize reflections. Buffers and Transceivers with  $I_{ol}=64\text{mA}$  current sink capability will be used. A split address buffering scheme is used to distribute the capacitive loading due to the large number of memory devices on the address bus. 74F657 Octal Bidirectional Transceivers with 8-Bit Parity Generator/Checker are used to buffer the 68020 Data Bus to and from the Memory Array. Only the DPA ROM memory is connected directly to the 68020 data and address bus.

### 10.2 MEMORY SUBSYSTEM CONTROL

Logic within the AMI will provide control signals such as WE\_L, CS\_L, OE\_L for the SRAM, EEPROM and DPA ROM. The control signals for the memory subsystem are driven directly from the AMI Gate Array with no external buffering to maintain the fastest possible control path. The AMI provides direction and enable control for the 74F657 Data Bus Transceivers.

### 10.3 SRAM

- Density = 256KByte configured in (8)32KX9 SRAM devices for 32-Bits of Data and 4 Bits for Byte Parity.

### 10.4 EEPROM

- Density = 256KByte configured in (8)32KX8 EEPROMs for 32-Bits of Data.

### 10.5 DPA PROM

- Density = 32Byte PROM justified to the 68020's least significant Byte D<31:24>

### 10.6 SRAM PARITY

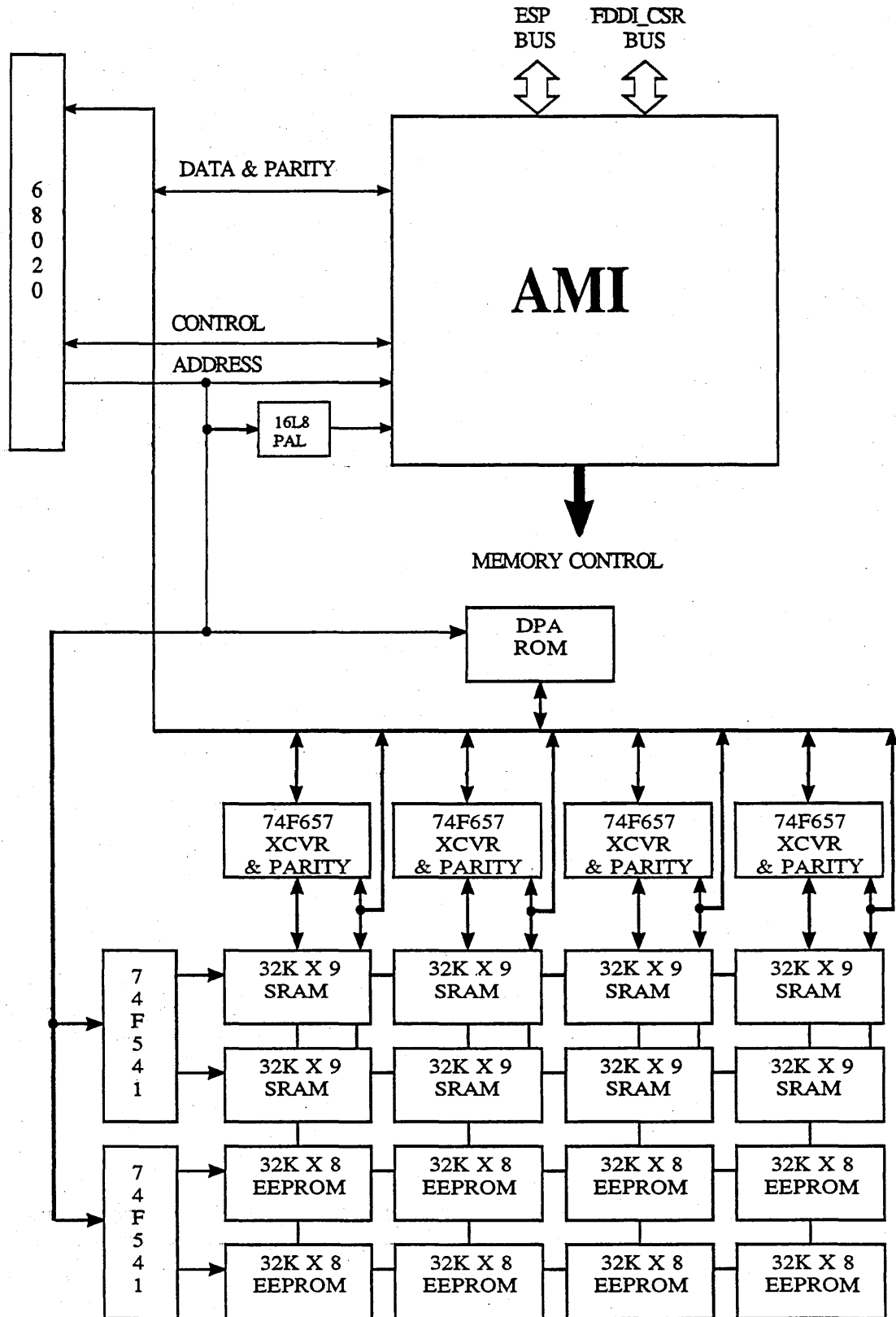
SRAM Parity is generated by logic within the 74F657 Transceivers. The SRAM used in the subsystem are 32Kx9 configuration and each SRAM device contains a parity bit for each corresponding byte. A parity bit is generated based on each byte of data from the 68020 and is stored in the 9th bit in each SRAM. On a 68020 SRAM read cycle, Parity is checked by logic internal to the AMI. Parity Bits are connected to the SRAM, 74F657 and the AMI.



## 10.7 SRAM & EEPROM READDRESSING

256KByte of SRAM and 256KByte of EEPROM will be installed in the memory subsystem for the first DEMFA. Provisions have been made in the control and address decode logic in the AMI to support up to 1MByte of SRAM and 1MByte of EEPROM. Memory upgrade changes can be made in the future and still make use of the AMI Gate Array design. Because a larger address space has been decoded than is actually installed, a word of caution is in order. Re-addressing of the installed SRAM and EEPROM will occur in each of the (4) 256KByte blocks of SRAM and EEPROM up to the total 1MByte of space provided for each.

Figure 20: 68020 Memory Subsystem Block Diagram



# CHAPTER 11

## PARITY LOGIC

### 11.1 OVERVIEW

During an SRAM memory write, a parity bit, generated by the 74F657 is written to the SRAM along with the data word for storage. When a read cycle occurs, parity checking is performed on the data word along with the original parity bit read from the SRAM memory. This parity checking is performed in the AMI gate array and odd parity is the default. Two physically separate pieces of logic are required for Parity Generation and Checking due to Gate Array I/O timing constraints. This logic will generate and check Byte Parity for the SRAMs in the 68020 Memory Subsystem. The 32KX9 configuration of SRAMs are used to provide storage for each byte of data along with its corresponding parity bit.

Three events are triggered when an SRAM parity error is detected; an Error Code is generated and sent to the ESP on the ERCOD<1:0> lines, the 68020 is set into RESET, and the 68020 Address at which the Parity Error occurred will be latched in the Parity Error Address Register (PEAR) and will survive a "warm" system reset. The Parity Error Address can be read by the 68020 after it wakes up again. "Warm Reset" is defined as an INIT condition without losing +5V DC. The 68020 will remain in the reset state until the INITL line is toggled.

During power up SRAM parity errors will not RESET the 68020. Bit<1> of the GPCSR must be set to a one in order to fully activate the parity error logic. Setting this bit will allow the logic to RESET the 68020 on an error.

A Self-Test mode is provided for testing the SRAM parity logic. The logic can be forced to check odd parity for individual bites and status can be obtained for these bites. For a detailed bit description refer to the GPCSR bit definitions (bit<9:1>) in CHAPTER 15.

Figure 21: PARITY CIRCUIT DIAGRAM

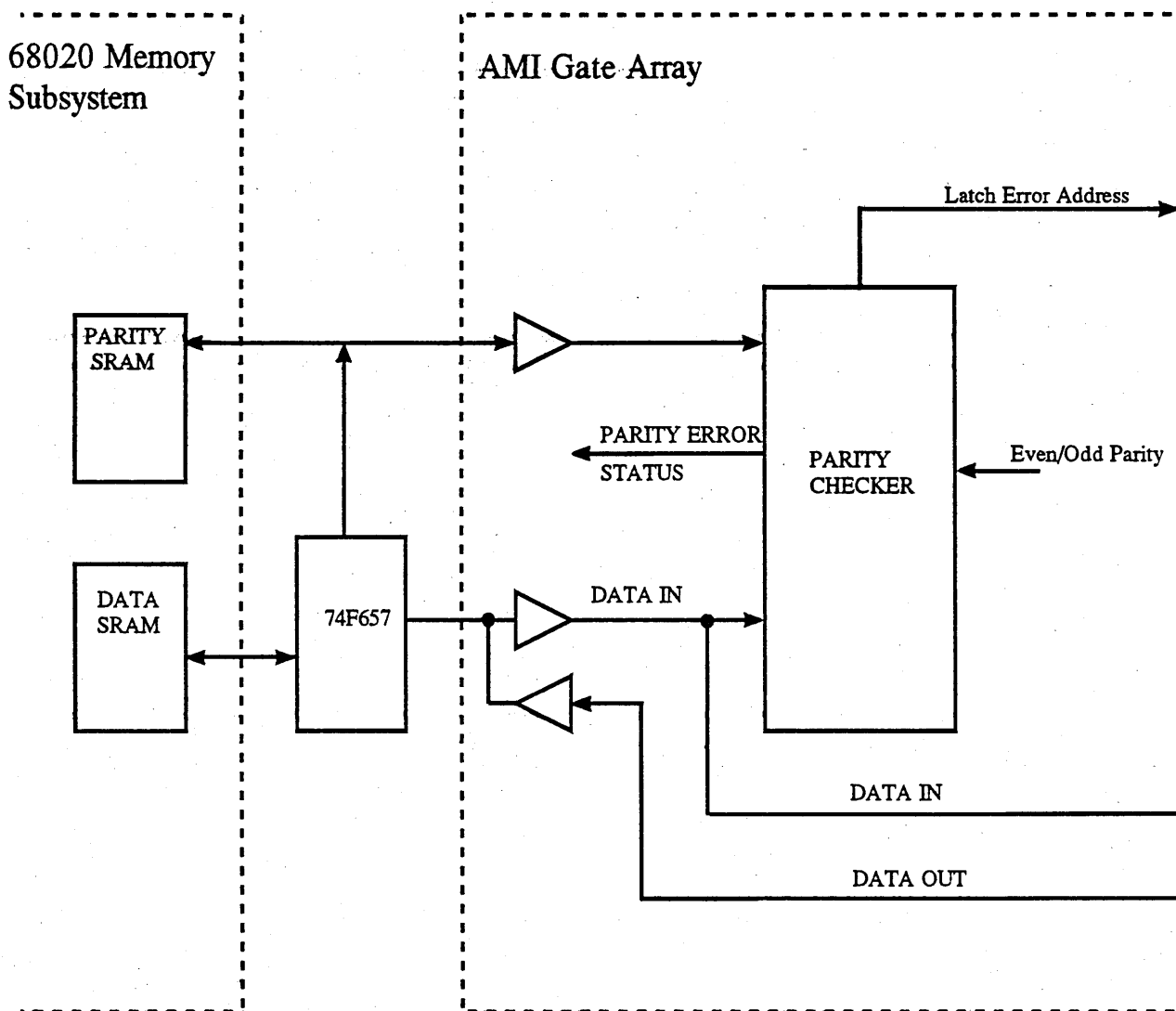
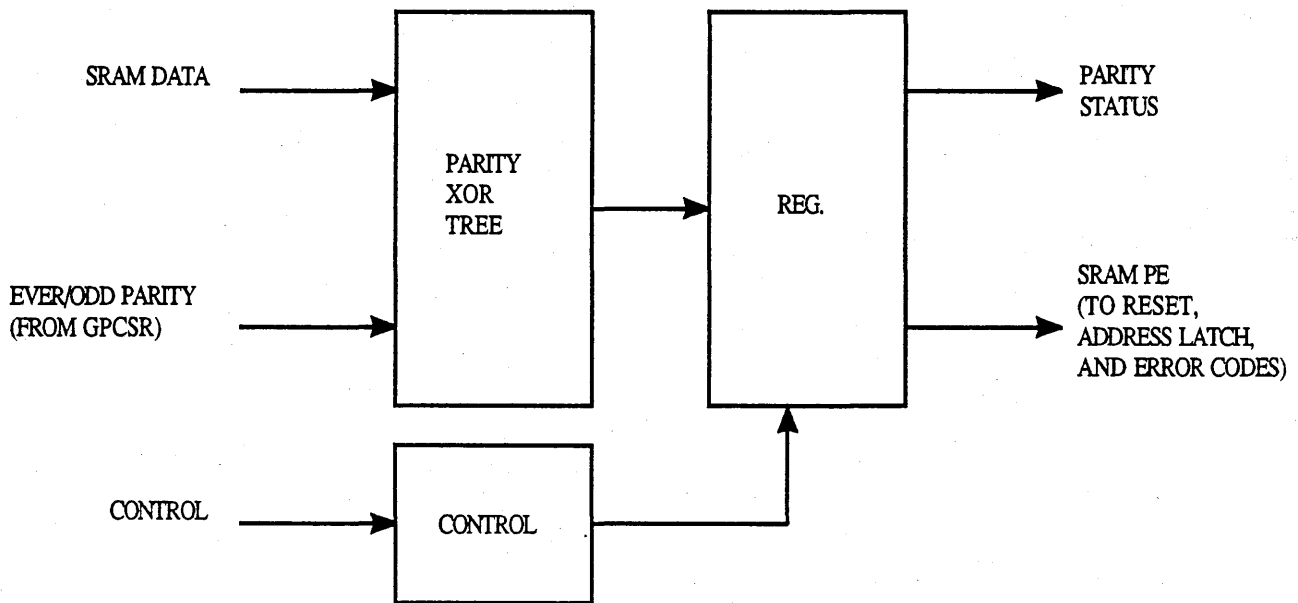


Figure 22: PARITY CIRCUIT DIAGRAM

# AMI\_SPAR



NOTE: PARITY GENERATED AND WRITTEN TO SRAM THRU USE OF 74F657.

# CHAPTER 12

## SELF-TEST LOGIC

### 12.1 OVERVIEW

This logic is required by the 68020 to help test the functionality and integrity of the AMI Gate Array and the 68020 Subsystem. The Self Test Logic is distributed throughout the AMI as needed. This includes logic to break up the Counters and Timers so they can be tested in smaller blocks, instead of waiting seconds for them to expire. Register read-back is employed where possible.

CSR accesses will be the method of setting up test modes.

#### 12.1.1 Memory Testing

##### 12.1.1.1 Local SRAM

Portions of 68020 Local SRAM must be tested at power up before executing further Self Test Code. Due to Self Test time constraints ( 9.899 seconds ), Self Test Code must be run out of SRAM. At power up, the Self Test must test SRAM and then move the Self Test Code from EEPROM into SRAM to allow code to execute with much better performance. This is necessary to enable a sufficient level of testing within the XMI allocated 10 seconds from init.

In actuality, the 10 second time also must include the time to start up RTOS which is estimated at 2 seconds and the 101msec of time needed to properly reset the 68020 on a cold start.

Due to this Self Test time constraints, the testing of Packet Memory requires a dedicated DRAM test logic in the PMC.

No special hardware is provided for testing Local SRAM.

##### 12.1.1.2 Local EEPROM

The contents of EEPROM will be verified at power up. A checksum will be calculated on the code as it is copied from EEPROM to SRAM. A separate checksum will be calculated for each major block of code: such as Self Test Code and Functional Code. The checksum calculated is compared against the value stored in EEPROM.

##### 12.1.1.3 Local DPA

The contents of the Node Address ROM will be verified at power up by reading all locations and calculating a checksum according to established procedures for Ethernet LAN Address ROMs. The checksum calculated is compared against the value stored in EEPROM.

#### 12.1.1.4 Packet Memory Testing

Packet Memory will be tested by a hardware assist in the PMC Gate Array. Further details of this hardware support are detailed in the PMC Design Spec.

A Self Test Mode will be provided in the PMC and AMI to allow the AMSTX bit from the PMC to be used as a Test Bit for timing the PMC DRAM Refresh Counters. The normal path of the AMSTX to the AMI Interrupt Logic can be disabled during this test and the interrupt polling method used. Also some masking must be done in the PMC - details can be found in the PMC Design Spec.

#### 12.1.2 Parity Logic Testing

A GPCSR Bit is provided to disable the SRAM Parity Error from Resetting the 68020. The GPCSR has ability to set ODD or EVEN Parity and Force Parity Errors to individual bytes. Byte Parity Status bits are also provided in the GPCSR. P>A 32-Bit latch is provided to latch the Address Bus contents on a parity error read cycle. The contents of this register can be read by the 68020. This register will only be initialized to all ones on a true power up cycle with XCI\_AC\_LO transitioning. This will allow it to be read on a warm start such as a Node Reset.

#### 12.1.3 Hardware & Watchdog Timers

These timers will be tested by chaining stages together to reduce the time required to test. The timers will be programmable and for test purposes will be readable by the 68020.

#### 12.1.4 AMI/ESP Interface

Currently the method of testing this block will be through 68020 writing and reading registers and checking data validity.

#### 12.1.5 CSR Bus Interface

The method of testing this block is through writing and reading registers and checking data validity.

#### 12.1.6 Interrupt Logic

Interrupts will be tested through their normal operating mode or through the use of the Interrupt Self Test Register (INTST). To improve test time we will allow polling of interrupts to determine correctness of response.

#### 12.1.7 DSACK & BUS ERROR Logic

Bus Error Logic including the Bus Error Timer is tested by programming the timer to a value and then performing an access to a non-decoded address ( NXM ). If the code TRAPS to the Bus Error Routine, then it worked successfully.

#### **12.1.8 Address Decode**

No specific test method except for writes and reads of various addressable function blocks.

#### **12.1.9 Reset Logic**

No test strategy as present time! If the processor wakes up properly and begins executing code, then the reset worked. A test is performed that tests the operation of the Reset Disable Bit in the GPCSR to insure that all external hardware gets reset but not the 68020.

#### **12.1.10 Memory Subsystem Control Logic**

There is no explicit testing of the Memory Subsystem Control Logic, but it is tested through all the other memory testing that is performed.



# CHAPTER 13

## LOGIC SUPPORTING ERROR CONDITIONS

### 13.1 OVERVIEW

This block will contain logic required to assist the 68020 or Adapter in the execution of a Error Response.

#### 13.1.1 AMI/ESP ERROR STATUS LINES

Two dedicated signal lines are connected between the AMI Gate Array and the ESP Gate Array for the purposes of signaling error conditions.

- **XERCOD[1] H:** This signal asserted alone signifies that an SRAM Parity Error has occurred, indicating a Fatal Error in the 68020 Subsystem. When this signal is asserted along with XERCOD[0] H it signifies a failure on the AMI/ESP Bus, also a Fatal Error.
- **XERCOD[0] H:** This signal asserted alone signifies that the Watchdog Timer has expired, indicating a Fatal Error in the 68020 Subsystem. When this signal is asserted along with XERCOD[1] H it signifies a failure on the AMI/ESP Bus, also a Fatal Error.

| XERCOD<1> | XERCOD<0> | Condition                | Action Taken by Host |
|-----------|-----------|--------------------------|----------------------|
| 0         | 0         | No Error                 | No Action            |
| 1         | 0         | SRAM Parity Error        | Issue Node Reset     |
| 0         | 1         | Watchdog Timer Expired   | Issue Node Reset     |
| 1         | 1         | AMI/ESP Bus Parity Error | Issue Node Reset     |

#### 13.1.2 BUS ERROR CONDITIONS

68020 accesses of Non-existent Memory will not generate a Data Acknowledge Control Signal and will thus cause the Bus Error Counter to Time out and generate a 68020 Bus Error Exception. In addition to a NXM access, an accesses to a Co-processor or Access Level Control Cycles will also time out with a Bus Error.

##### 13.1.2.1 IMPROPER ADDRESS ALIGNMENT

If the addressing rules set forth in this specification are violated, unpredictable results will be obtained on 68020 Read and Write Cycles. Strict adherence to longword, word or byte addressing details of the different function blocks within the AMI is crucial to proper operation.



# CHAPTER 14

## CONTROL STATUS REGISTERS & MISC. GLUE LOGIC

### 14.1 OVERVIEW

I/O logic is provided to essentially get signals in and out of the AMI Gate Array. In addition logic will be provided to support a scan method of testing I/O Cells and bond wires through the use of a NAND Tree. This logic also provides the ability to tri-state all external pins on the 68S Gate Array.

#### 14.1.1 CLOCKS INTERFACE

Three external 64ns clocks will be applied to this section that have their origins in the XMI Interface. One clock will be passed through a precision 10.7nsec delay line prior to entering the AMI to be used in generating a tightly controlled 50% duty cycle clock for the 68020 microprocessor.

The 64ns clocks will also be used to sequence the internal control logic. This logic will also drive and distribute the clock throughout the gate array.

Clocks XCI\_34H and XCI\_61H will be required to synchronize the ESP Bus Interface logic.

Another clock called BYTCLK will be used in the FDDI Chips CSR Bus Interface to synchronize that logic to the 80nsec clock used by devices on the CSR bus.

#### 14.1.2 AMI CSRs

A catch all General Purpose Control Status Register is provided, the details of the register functions are provided in the AMI Registers section of this specification.

#### 14.1.3 SPECIAL CSR ACCESSES

To generate the following events, 68020 software must perform a write to the specified CSR address for each function. The hardware will be designed to generate a strobe when a write to the specified CSR address occurs. The data written is not important, since it is the act of writing to the address that causes the event to occur. For definition purposes, a write of all 0's to the CSR will suffice.

- **AMXDN:** Adapter Manager Transmit Done signal is generated through a CSR access written by the 68020 when it wants to tell the PMC that it has taken a packet off of the AM Transmit Ring. This signal must be asserted after every packet transaction of this type.

This signal is synchronized to the 80nsec BYTCLK and produces a single 80nsec pulse on the AMXDN signal line when the defined CSR is accessed by the 68020.

- **AMRDN:** Adapter Manager Receive Done signal is generated through a CSR access written by the 68020 when it wants to tell the PMC that it has put a packet onto the AM Receive Ring. This signal must be asserted after every packet transaction of this type.

This signal is synchronized to the 80nsec BYTCLK and produces a single 80nsec pulse on the AMRDN signal line when the defined CSR is accessed by the 68020.

- **SMT\_PRIORITY\_H:** This is a software generated interrupt that is required by RTOS to assist in pre-empting SMT code execution. This strobe will be in synch with the 64nsec XCI\_34H Clock. This pulse is supplied to the Interrupt Control logic to generate the interrupt. This interrupt is cleared on an IACK cycle.
- **RTOS\_EVENT\_H:** This is a software generated interrupt that is required by RTOS to assist in task scheduling. This strobe will be in synch with the 64nsec XCI\_34H Clock. This pulse is supplied to the Interrupt Control logic to generate the interrupt. This interrupt is cleared on an IACK cycle.

# CHAPTER 15

## UPDATE (02-19-1990) AMI REGISTERS

This chapter describes all the registers available within the AMI gate array.

### Conventions:

All address values stated in hex format. Byte, word and longword addressing conforms to the Motorola big-endian Protocol ( Byte 0 is always the most significant D[31:24] byte ).

### Dynamic Bus Sizing

The MC68020 dynamically interprets the port size of the addressed register within the AMI Gate Array during each cycle, allowing transfers to or from 8-, 16-, and 32-bit ports. During a transfer cycle, the slave device signals its port size (byte, word, or longword) and indicates completion of the bus cycle to the processor through the use of the DSACKL[1:0] signal lines.

For example if the 68020 is executing an instruction that reads a longword from a longword aligned address, it attempts to read 32 bits during the first bus cycle. If the port responds that it is 16 bits wide, the 68020 latches 16 bits of valid data ( D[31:16] ) and executes another bus cycle to obtain the remaining 16 bits ( again on D[31:16] ). For 16-bit word reads, the AMI drives D[15:0] with all zeros.

### 15.1 ESP INTERFACE STATUS REGISTER (ESPFSR)

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

Adapter Manager ( 68020 ) Physical Address: \$0010 0A00  
=====

Access: RO ( Read Only )  
=====

Initialized as: All bits zero.  
=====

```
 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|   |H|P|
|   |B|B|
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Bit Descriptions:  
=====

BIT<31:02>: NIO ( Not Implemented, Read as zero )

BIT<01>: Host/XMI Busy; 1 = Busy, 0 = Not Busy  
This status bit when set informs the 68020 that an ESP to Host Transaction is in progress. This bit will be set by the ESP G.A. when the 68020 writes the AMXMIH Register. While this bit is set, the 68020 must not initiate a new transaction by writing the AMXMIH Register again. Once the bit is cleared, a new transaction can be initiated.

BIT<00>: PMC Busy; 1 = Busy, 0 = Not Busy  
This status bit when set informs the 68020 that an ESP to PMC Transaction is in progress. This bit will be set by the ESP G.A. when the 68020 writes the AMPBAR Register. While this bit is set, the 68020 must not initiate a new transaction by writing the AMPBAR Register again. Once the bit is cleared, a new transaction can be initiated.

## 15.2 TIMERS AND COUNTERS

### 15.2.1 AMI WATCHDOG TIMER (WDTIMR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C00

Access: R/W ( Read / Write )

Initialized as: All bits zero.

```

3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                               Counter Preset Value                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

BIT<31:00>: Data to be loaded into the Watchdog Counter as a preset value. The counter used for the Watchdog Timer is 32-Bits wide. Bit<31> is the Most Significant Counter Bit and Bit<0> is the Least Significant Counter bit. Each binary bit weight is multiplied times the 64nsec clock period to calculate the counter expiration time. The counter will decrement from this preset value and interrupt the 68020 when it reaches \$0000 0000.

A readback of this register will provide the latest counter value at the time of the 68020 read cycle.

This preset value is reloaded into the counter and a new count down sequence will start whenever the Watchdog Timer is cleared by software.

### 15.2.2 AMI RTOS TIMER (RTOSTIMR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C04

Access: R/W ( Read / Write )

Initialized as: All bits zero.

```

3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|0|0|0|0|0|0|0|0|0|                               Counter Preset Value                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

BIT<31:24>: MBZ ( Must Be Zero )

BIT<23:00>: Data to be loaded into the RTOS Counter as a preset value. The counter used for the RTOS Timer is 24-Bits wide allowing an elapsed time of up to 1 second before initiating a 68020 RTOS Timer Interrupt.

Bit<23> is the Most Significant Counter Bit and Bit<0> is the Least Significant Counter Bit. Each binary bit weight is multiplied times the 64nsec clock period to calculate the counter expiration time. The counter will decrement from this preset value on each clock cycle and generate an interrupt to the 68020 when it reaches \$0000 0000. Once enabled, this Counter will free run by counting down, reloading the initial counter value and repeating the count down sequence.

A readback of this register will provide the latest counter value at the time of the 68020 read cycle.

### 15.2.3 COUNTER/TIMER CSR (CTCSR)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C08

Access: R/W ( Read / Write )

Initialized as: All bits zero.

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +-----+-----+-----+-----+-----+-----+-----+-----+
	B	R	W	B	R	W
	E	T	D	E	T	D
	C	C	S	T	T	T
	T	T	T	E	E	E
  +-----+-----+-----+-----+-----+-----+-----+-----+
  
```

Bit Descriptions:

- BIT<31:06>: NIO ( Not implemented, Read as zero )
- BIT<05>: Bus Error Counter Self Test Mode; 1 = Enabled, 0 = Disabled  
 This bit is for Bus Error Counter Self Test. It allows one to test the operation of the Bus Error Timer by chaining the counter stages together to minimize test time. When this bit is at a logic "1", the Bus Error Counters will be chained together. When set to the default logic "0", the counters will be set for normal operation. This bit is Read/Write access.
- BIT<04>: RTOS Counter Self Test Mode; 1 = Enabled, 0 = Disabled  
 This bit is for RTOS Counter Self Test. It allows one to test the operation of the RTOS Timer by chaining the counter stages together to minimize test time. When this bit is at a logic "1", the RTOS Counters will be chained together. When set to the default logic "0", the counters will be set for normal operation. This bit is Read/Write access.
- BIT<03>: Watchdog Timer Self Test Mode; 1 = Enabled, 0 = Disabled  
 This bit is for Self Test Code to test the operation of the Watchdog Timer by chaining the counter stages together to minimize test time. When this bit is at a logic "1", the Watchdog Counters will be chained together. When set to the default logic "0", the counters will be set for normal operation. This bit is Read/Write access.
- BIT<02>: Bus Error Timer Enable; 1 = Enable, 0 = Disable.  
 When set to logic "1" this bit will enable the Bus Error Counter to be operational. The value of BET Register should be loaded prior to setting this bit. The default value of this bit at RESET is zero for counter disabled. This bit is Read/Write Access.  
  
 Once enabled, the counter will start counting at the beginning of a 68020 Bus Cycle and be reloaded at the end of a 68020 Bus Cycle. When set to a logic "0" this bit will disable all Bus Error Counter operation.
- BIT<01>: RTOS Timer Enable; 1 = Enable, 0 = Disable.  
 When set to logic "1" this bit will enable the RTOS Counter to be operational. The value of RTOSTIMR Register should be loaded prior to setting this bit. The default value of this bit at RESET is zero. This bit is Read/Write Access.



The counter will start counting once enabled and free run. When set to a logic "0", this bit will disable RTOS Counter operation.

BIT<00>: Watchdog Timer Enable; 1 = Enable, 0 = Disable.

When set to logic "1" this bit will enable the Watchdog Counter to be operational. The value of WDTIMR Register should be loaded prior to setting this bit. The default value of this bit at RESET is zero for counter disabled. This bit is Read/Write Access.

Once enabled, the counter will start counting down. In normal operation, software will clear the counter before it reaches a count of \$0000 0000 and issues a Watchdog Timer Interrupt. The Watchdog Timer is cleared and reloaded with a write to WDTCLR Register.

### 15.2.4 BUS ERROR TIMER (BET)

Adapter Manager ( 68020 ) Physical Address: \$0010 0C10

Access: R/W ( Read / Write )

Initialized as: \$0000 00FF ( approx 16usec. )

```

3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
|                               Counter Preset                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Bit Descriptions:

BIT<31:16>: MBZ ( Must Be Zero )

BIT<15:00>: Data to be loaded into the Bus Error Counter as a preset value. The counter used for the Bus Error Timer is 16-Bits wide allowing an elapsed time of up to 4 msec. before initiating a 68020 Bus Error Cycle.

Bit<15> is the Most Significant Counter Bit and Bit<0> is the Least Significant Counter Bit. Each binary bit weight is multiplied times the 64nsec clock period to calculate the counter expiration time. The counter will decrement from this preset value on each clock cycle and generate a Bus Error to the 68020 when it reaches \$0000 0000. The Bus Error Counter will be enabled and start counting whenever a new 68020 Bus Cycle starts. The counter will be stopped and reloaded when DSACKL[1:0] is asserted. The counter is reloaded to the value written in the register above at the normal completion of a 68020 Bus Cycle.

A readback of this register will provide the latest counter value at the time of the 68020 read cycle. A default value for the register other than zero is provided to assist Self Test Code.

### 15.3 AMI GENERAL PURPOSE CSR (GPCSR)

Adapter Manager ( 68020 ) Physical Address: \$0010 1000

Access: SC ( Special Case - see bit descriptions )

Initialized as: All bits zero.

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
	E	R	R	R	R	N	N	N	N	C	C	R	R	S	R	T	E	C	S	S	S	S	S	S	S	S	C
	E	M	M	M	P	P	P	P	L	L	T	T	T	S	E	S	L	R	R	R	R	R	R	R	R	R	D
	W	S	S	S	E	E	S	S	R	R	E	E	S	T	S	P	R	S	S	S	S	E	E	E	D	I	
	E	L	L	L	1	0	T	T	E	B	S	S	E	D	T	E	O	T	T	T	T	3	2	1	0	I	S
	N	2	1	0			1	0			1	0	L		S			3	2	1	0				S		
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Bit Descriptions:

BIT<31:27>: NIO ( Not implemented, Read as zero )

BIT<26>: EEPROM Write Enable, 1 = Enabled, 0 = Disabled  
 This bit enables the EEPROM Write Strobe to function. Its purpose is as a write protect against un-intended EEPROM Write Accesses. With this bit set to "0" or disabled, a write to EEPROM will appear to function correctly with the exception of the EEWEL control signal to the EEPROMs not asserting and de-asserting. The current cycle's data on the EEPROM Data Bus will not be written into the EEPROM. For firmware to perform writes to EEPROM, code must first enable this bit before writing to EEPROM. At reset, this bit defaults to a "0" or Function Disabled. This bit is READ/WRITE accessible.

BIT<25:23>: Reset Mux Select <2:0>, 1 = Enabled, 0 = Disabled  
 These bits are for use in Gate Array Fault Grading and should NOT be programmed during Self Test or Normal Operation. The default value will be read as all 0's. The following shows mux select codes vrs. load data for the Reset Counter. These values are only loaded into the Reset Counter following the cycling of INIT L provided RTESTO ( GPCSR bit <15> ) is set to "1". At powerup, the bits are set to "0" selecting a reset counter timeout of 101msec.

| RMSL<2:0> | Reset Counter Load Value |
|-----------|--------------------------|
| 000       | \$18 148E ( 101msec. )   |
| 001       | \$1F FFFF ( 134msec. )   |
| 010       | \$01 0000 ( 4.2msec. )   |
| 011       | \$00 0100 ( 16usec. )    |
| 100       | \$00 0001 ( 64nsec. )    |
| 101       | \$00 0007 ( 512nsec. )   |
| 110       | \$00 0230 ( 35usec. )    |
| 111       | \$00 0007 ( 512nsec. )   |

BIT<22>: NP (CSR) Bus Even Parity, BYTE 1; 1 = Enabled, 0 = Disabled  
 This bit, when set to logic "1", selects EVEN Parity checking for NP (CSR) Bus Byte 1 Parity. The default value of this bit at logic "0" selects ODD Parity. This bit is Read/Write Access.  
 This bit is for testing the Parity Protection Path between the PARSER and the AMI.

Digital Equipment Corporation - VAX Products and Options  
Confidential and Proprietary

- BIT<21>: NP (CSR) Bus Even Parity, BYTE 0; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for NP (CSR) Bus Byte 0 Parity. The default value of this bit at logic "0" selects ODD Parity. This bit is Read/Write Access.  
This bit is for testing the Parity Protection Path between the PARSER and the AMI.
- BIT<20>: NP (CSR) Bus Parity Error BYTE 1; 1 = Error, 0 = No Error  
This is a Parity Error Status Bit for Byte 1 or NPD[7:0] of the NP (CSR) Bus. This bit provides parity status to be used by Self Test Code to test Byte Parity. This status bit is read only and defaults to "0" on RESET.
- BIT<19>: NP (CSR) Bus Parity Error BYTE 0; 1 = Error, 0 = No Error  
This is a Parity Error Status Bit for Byte 0 or NPD[15:8] of the NP (CSR) Bus. This bit provides parity status to be used by Self Test Code to test Byte Parity. This status bit is read only and defaults to "0" on RESET.
- BIT<18>: Clear Error Codes;  
Writing a 1 to this bit will generate a pulse that will clear the error codes sent to the ESP gate-array. This bit is write 1 only and read as 0. This bit defaults to "0" on RESET.
- BIT<17>: Clear Busy;  
Writing a 1 to this bit will generate a pulse that will clear the Busy Bits from the ESP gate-array. During Self Test, register accesses to the ESP Registers will not follow the ESP Bus Protocol and the ESP will not cycle the XMI Busy or PMC Busy bits correctly. This bit allows clearing out the AMI/ESP Interface Logic that is expecting the busy bits to be cycled correctly at this time. This bit is write 1 only and read as 0. This bit defaults to "0" on RESET.
- BIT<16>: RTEST1; 1 = Enabled, 0 = Disabled  
Reset Test 1 is a bit that when set enables the testing of the AMI Reset Logic during Fault Grading. This bit when set to a logic "1" will gang stages of the Reset Counter together for faster testing. This bit powers up in the logic "0" state or Disabled. This bit serves no purpose during Self Test or Normal Operation but is an assist to test during fault grading. This bit should never be set and will always read as a 0.
- BIT<15>: RTEST0; 1 = Enabled, 0 = Disabled  
Reset Test 0 is a bit that when set enables the testing of the AMI Reset Logic during Fault Grading. This bit when set to a logic "1" will block the INIT L signal from resetting the Reset Counter Mux Select Logic from its loaded value. This bit powers up in the logic "0" state or Disabled. This bit serves no purpose during Self Test or Normal Operation but is an assist to test during Fault Grading. This bit should never be set and will always read as a 0.
- BIT<14>: Self-Test Select; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", Enables the use of the Interrupt self-test register to generate interrupts. This bit is for testing the interrupt logic in the AMI gate-array. This bit is read/write and defaults to "0" on RESET.
- BIT<13>: Reset Disable; 1 = Enabled, 0 = Disabled  
This bit, when set to a logic "1", Enables the logic that WILL PREVENT a Node Reset from propagating through the AMI and resetting the 68020. This function is provided for the "warm resetting" of the Adapter. A time delay after the INIT L signal has been pulsed, logic will auto-clear this bit to logic "0" and Disable the Function.

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

- BIT<12>:** TEST\_SEL H or Test Select; 1 = FDDI CSR Intfc, 0 = ESP Intfc Control  
This bit, is used to control a mux that gates buried control signals out of the AMI for testing purposes. The default value of this bit is logic "0" and selects ESP Control Signals. When set to a logic "1", it selects FDDI CSR BUS State Machine Bits. The control signals from the mux are brought out on TEST3,TEST2,TEST1,TEST0. This bit is Read/Write Access. This bit is used primarily for Fault Grading of the AMI.
- BIT<11>:** ESP/AMI Interface Bus Even Parity Check; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking on AMI/ESP Bus reads. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access. This bit is for testing the Parity Protection Path on the AMI/ESP Bus.  
  
AMI/ESP Bus Parity Generation is always ODD Parity.
- BIT<10>:** Clear Ownership; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1" will cause an automatic Clear Buffer Ownership to occur after a Buffer Descriptor Write in the AMI/ESP Interface's PM Direct Addressed mode of operation.  
  
When this bit is set to logic "0", no clear ownership will occur after a Buffer Descriptor Write. Logic "0" is the default setting of the bit at RESET. This bit is Read/Write.
- BIT<09>:** SRAM Parity Error Byte 3; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 3 or D[7:0] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 3 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.
- BIT<08>:** SRAM Parity Error Byte 2; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 2 or D[15:8] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 2 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.
- BIT<07>:** SRAM Parity Error Byte 1; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 1 or D[23:15] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 1 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.
- BIT<06>:** SRAM Parity Error Byte 0; 1 = Error, 0 = No error  
This is a Parity Error Status Bit for Byte 0 or D[31:23] of SRAM. This bit provides parity status to be used by Self Test Code to test Byte Parity. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 0, SRAM Parity Error Byte 0 can be written with a 1 which will clear this Parity Error status bit and the Parity Error Address Register. With the SRAM Parity Reset Disable (BIT<01> of GPCSR) set to a 1 this status bit is read only and defaults to "0" on RESET.
- BIT<05>:** SRAM Even Parity Byte 3; 1 = Enabled, 0 = Disabled  
This bit, when set to logic "1", selects EVEN Parity checking for Byte 3 or D[7:0] of SRAM. The default value of this bit at logic "0" selects ODD Parity checking. This bit is Read/Write Access.



BIT<31:00>: MBZ ( Must Be Written as Zero )  
A longword write of any data to this address will generate the Adapter Manager Transmit Done ( AMXDN ) strobe. AMXDN is a signal to the PMC Gate Array that notifies it that the Adapter Manager has taken a packet off the Adapter Manager's Transmit Ring. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

### 15.4.2 ADAPTER MANAGER RECEIVE DONE STROBE (AMRDS)

Adapter Manager ( 68020 ) Physical Address: \$0010 1014  
=====

Access: SC ( Special Case - see bit descriptions )  
=====

Initialized as: Not Applicable  
=====

```
 3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+-----+-----+-----+-----+-----+-----+-----+-----+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
+-----+-----+-----+-----+-----+-----+-----+-----+
```

Bit Descriptions:  
=====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
A longword write of any data to this address will generate the Adapter Manager Receive Done ( AMRDN ) strobe. AMRDN is a signal to the PMC Gate Array that notifies it that the Adapter Manager has put a packet on the Adapter Manager's Receive Ring. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

### 15.4.3 SMT PRIORITY STROBE (SMTPS)

Adapter Manager ( 68020 ) Physical Address: \$0010 1018  
=====

Access: SC ( Special Case - see bit descriptions )  
=====

Initialized as: Not Applicable  
=====

```
 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+--+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
+--+
```

Bit Descriptions:  
=====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
A longword write of any data to this address will generate the SMT\_PRIORITY interrupt. This strobe is used by SMT Code to enable the pre-emption of SMT Processing by generating an interrupt. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

#### 15.4.4 RTOS EVENT STROBE (RTOSES)

Adapter Manager ( 68020 ) Physical Address: \$0010 101C  
=====

Access: SC ( Special Case - see bit descriptions )  
=====

Initialized as: Not Applicable  
=====

```
 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0
 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
+--+
|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
+--+
```

Bit Descriptions:  
=====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
A longword write of any data to this address will generate the RTOS\_EVENT interrupt. This interrupt is used by RTOS to assist in the scheduling of tasks. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

#### 15.4.5 WATCHDOG TIMER CLEAR STROBE (WDTCLR)

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

Adapter Manager ( 68020 ) Physical Address: \$0010 0C0C  
 =====

Access: SC ( Special Case - see bit descriptions )  
 =====

Initialized as: Not Applicable  
 =====

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

**Bit Descriptions:**  
 =====

BIT<31:00>: MBZ ( Must Be Written as Zero )  
 A longword write of any data to this address will generate the Watchdog Timer Clear. This strobe is used by the Watchdog Counter logic to clear the current count and reload the value of WDTIMR Register into the counter. The act of writing to this address decodes the strobe, the action is independent of the data. This is Write Only Access. Since there is no actual register, there is no default bit setting.

**15.5 AMI MISCELANEOUS REGISTERS**

**15.5.1 PARITY ERROR ADDRESS REGISTER (PEAR)**

Adapter Manager ( 68020 ) Physical Address: \$0010 1200  
 =====

Access: RO ( Read Only )  
 =====

INITIALIZED AS: \$00FF FFFF for power up reset.  
 =====

```

  3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  |0|0|0|0|0|0|0|0|          PE ADDRESS          |
  +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  
```

BIT<31:24>: NIO ( Not implemented, read as zero )  
 BIT<23:00>: Unless all set to logic "high", this longword will contain the SRAM memory address of the last SRAM Parity Error to occur.

Note: To clear this register SRAM Parity Reset Disable (BIT<01> of GPCSR) must be set to a 0, and one of the SRAM Parity Error Byte bits (BIT<09:06> of GPCSR) must be written to a 1.

**15.6 INTERRUPT AND INTERRUPT VECTOR REGISTERS**

**15.6.1 INTERRUPT CSR 1 (INTCSR1)**



Adapter Manager ( 68020 ) Physical Address: \$0010 1400  
 =====

Access: RO ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                                                                                                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------------------------------------------------------------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                                                                                                 | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |
| 1                                                                                                 | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| NIO NIO NIO NIO NIO IPK IPJ IPI IPH IPG IPF IPE IPD IPC IPB IPA                                   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

Bit Descriptions:  
 =====

- BIT[31:27]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[26]: IPK-RESERVED INTERRUPT PENDING  
 A 1 in this bit location indicates that a RESERVED interrupt is pending. This bit is used to poll the RESERVED interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[25]: IPJ-ESP CHIP INTERRUPT PENDING  
 A 1 in this bit location indicates that an ESP CHIP interrupt is pending. This bit is used to poll the ESP CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[24]: IPI-PM START TRANSMIT INTERRUPT PENDING  
 A 1 in this bit location indicates that an PM START TRANSMIT interrupt is pending. This bit is used to poll the PM START TRANSMIT interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[23]: IPH-FDDI NP CSR BUS PARITY ERROR INTERRUPT PENDING  
 A 1 in this bit location indicates that an FDDI NP CSR BUS PARITY ERROR interrupt is pending. This bit is used to poll the FDDI NP CSR BUS PARITY ERROR interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[22]: IPG-FDDI ELM CHIP INTERRUPT PENDING  
 A 1 in this bit location indicates that an FDDI ELM CHIP interrupt is pending. This bit is used to poll the FDDI ELM CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

- BIT[21]:** IPF-FDDI MAC CHIP INTERRUPT PENDING  
A 1 in this bit location indicates that an FDDI MAC CHIP interrupt is pending. This bit is used to poll the FDDI MAC CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[20]:** IPE-FDDI RMC CHIP INTERRUPT PENDING  
A 1 in this bit location indicates that an FDDI RMC CHIP interrupt is pending. This bit is used to poll the FDDI RMC CHIP interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[19]:** IPD-SMT PRIORITY INTERRUPT PENDING  
A 1 in this bit location indicates that an SMT PRIORITY interrupt is pending. This bit is used to poll the SMT PRIORITY interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[18]:** IPC-PARSER INTERRUPT PENDING  
A 1 in this bit location indicates that an PARSER interrupt is pending. This bit is used to poll the PARSER interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[17]:** IPB-HARDWARE TIMER INTERRUPT PENDING  
A 1 in this bit location indicates that an HARDWARE TIMER interrupt is pending. This bit is used to poll the HARDWARE TIMER interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.
- BIT[16]:** IPA-EVENT LEVEL MONITOR INTERRUPT PENDING  
A 1 in this bit location indicates that an EVENT LEVEL MONITOR interrupt is pending. This bit is used to poll the EVENT LEVEL MONITOR interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the SOURCE INTERRUPT MASK REGISTER. WHEN IN poll mode a 1 must be written in this bit to clear. In normal operation the IACK cycle will cause this bit to be cleared.

## **15.6.2 INTERRUPT CSR 2 (INTCSR2)**

Adapter Manager ( 68020 ) Physical Address: \$0010 1402  
 =====

Access: RO ( Read Only )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                                                                                                   |   |         |   |         |   |         |   |         |   |         |   |         |   |         |   |
|---------------------------------------------------------------------------------------------------|---|---------|---|---------|---|---------|---|---------|---|---------|---|---------|---|---------|---|
| 3                                                                                                 | 3 | 2       | 2 | 2       | 2 | 2       | 2 | 2       | 2 | 2       | 2 | 1       | 1 | 1       | 1 |
| 1                                                                                                 | 0 | 9       | 8 | 7       | 6 | 5       | 4 | 3       | 2 | 1       | 0 | 9       | 8 | 7       | 6 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |         |   |         |   |         |   |         |   |         |   |         |   |         |   |
| NIO NIO                                                                                           |   | NIO NIO |   | NIO NIO |   | NIO NIO |   | NIO NIO |   | IP6 IP5 |   | IP4 IP3 |   | IP2 IP1 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |         |   |         |   |         |   |         |   |         |   |         |   |         |   |

Bit Descriptions:  
 =====

BIT[31:22]: NIO-NOT IMPLEMENTED, READ AS 0

BIT[21]: IP6-IPL6 PENDING  
 A 1 in this bit location indicates that an IPL6 interrupt is pending. This bit is used to poll IPL6 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

BIT[20]: IP5-IPL5 PENDING  
 A 1 in this bit location indicates that an IPL5 interrupt is pending. This bit is used to poll IPL5 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

BIT[19]: IP4-IPL4 PENDING  
 A 1 in this bit location indicates that an IPL4 interrupt is pending. This bit is used to poll IPL4 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

BIT[18]: IP3-IPL3 PENDING  
 A 1 in this bit location indicates that an IPL3 interrupt is pending. This bit is used to poll IPL3 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

BIT[17]: IP2-IPL2 PENDING  
 A 1 in this bit location indicates that an IPL2 interrupt is pending. This bit is used to poll IPL2 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

BIT[16]: IP1-IPL1 PENDING  
 A 1 in this bit location indicates that an IPL1 interrupt is pending. This bit is used to poll IPL1 interrupt when it is being masked. To mask this interrupt the corresponding mask bit must be set in the IPL INTERRUPT MASK REGISTER.

NOTE: IPL7 IS NON-MASKABLE

### 15.6.3 SOURCE INTERRUPT MASK REGISTER (SIMR)

Adapter Manager ( 68020 ) Physical Address: \$0010 1404

Access: R/W ( Read / Write )

INITIALIZED AS: All bits zero.

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3   | 3   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 1   | 1   | 1   | 1   |     |
| 1   | 0   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   | 9   | 8   | 7   | 6   |
| NIO | NIO | NIO | NIO | NIO | RES | ESP | PMX | FCP | ELM | MAC | RMC | SMT | PAR | HWT | EVL |

Bit Descriptions:

BIT[31:27]: NIO-NOT IMPLEMENTED, READ AS 0

BIT[26]: RES-RESERVED  
 Writing a 1 to this bit location will mask RESERVED interrupts. Masking RESERVED will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

BIT[25]: ESP-MASK ESP CHIP INTERRUPT  
 Writing a 1 to this bit location will mask ESP CHIP interrupts. Masking ESP CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

BIT[24]: PMX-MASK PM START TRANSMIT INTERRUPT  
 Writing a 1 to this bit location will mask PM START TRANSMIT interrupts. Masking PM START TRANSMIT will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

BIT[23]: FCP-MASK FDDI CSR PARITY ERROR INTERRUPT  
 Writing a 1 to this bit location will mask FDDI CSR PARITY ERROR interrupts. Masking FDDI CSR PARITY ERROR will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

BIT[22]: ELM-MASK FDDI ELM CHIP INTERRUPT  
 Writing a 1 to this bit location will mask FDDI ELM CHIP interrupts. Masking FDDI ELM CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

BIT[21]: MAC-MASK FDDI MAC CHIP INTERRUPT  
 Writing a 1 to this bit location will mask FDDI MAC CHIP interrupts. Masking FDDI MAC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

- BIT[20]: RMC-MASK FDDI RMC CHIP INTERRUPT  
Writing a 1 to this bit location will mask FDDI RMC CHIP interrupts. Masking FDDI RMC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[19]: SMT-MASK SMT PRIORITY INTERRUPT  
Writing a 1 to this bit location will mask SMT PRIORITY interrupts. Masking SMT PRIORITY will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[18]: PAR-MASK PARSER INTERRUPT  
Writing a 1 to this bit location will mask PARSER interrupts. Masking PARSER will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[17]: HWT-MASK HARDWARE TIMER INTERRUPT  
Writing a 1 to this bit location will mask HARDWARE TIMER interrupts. Masking HARDWARE TIMER will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.
- BIT[16]: EVL-MASK EVENT LEVEL MONITOR INTERRUPT  
Writing a 1 to this bit location will mask EVENT LEVEL MONITOR interrupts. Masking EVENT LEVEL MONITOR will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. Writing a 0 to this bit will allow this interrupt to propagate out to the 68020.

#### 15.6.4 IPL INTERRUPT MASK REGISTER (IPLIMR)

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

Adapter Manager ( 68020 ) Physical Address: \$0010 1406  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                                                                                             |   |         |   |         |   |         |   |         |   |         |   |         |   |         |   |
|---------------------------------------------------------------------------------------------|---|---------|---|---------|---|---------|---|---------|---|---------|---|---------|---|---------|---|
| 3                                                                                           | 3 | 2       | 2 | 2       | 2 | 2       | 2 | 2       | 2 | 2       | 1 | 1       | 1 | 1       |   |
| 1                                                                                           | 0 | 9       | 8 | 7       | 6 | 5       | 4 | 3       | 2 | 1       | 0 | 9       | 8 | 7       | 6 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |         |   |         |   |         |   |         |   |         |   |         |   |         |   |
| NIO NIO                                                                                     |   | NIO NIO |   | NIO NIO |   | NIO NIO |   | NIO NIO |   | IM6 IM5 |   | IM4 IM3 |   | IM2 IM1 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |         |   |         |   |         |   |         |   |         |   |         |   |         |   |

**Bit Descriptions:**  
 =====

BIT[31:22]: NIO-NOT IMPLEMENTED, READ AS 0

BIT[21]: IM6-MASK IPL6  
 Writing a 1 to this bit location will mask IPL6 interrupts. Masking IPL6 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

BIT[20]: IM5-MASK IPL5  
 Writing a 1 to this bit location will mask IPL5 interrupts. Masking IPL5 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

BIT[19]: IM4-MASK IPL4  
 Writing a 1 to this bit location will mask IPL4 interrupts. Masking IPL4 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

BIT[18]: IM3-MASK IPL3  
 Writing a 1 to this bit location will mask IPL3 interrupts. Masking IPL3 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

BIT[17]: IM2-MASK IPL2  
 Writing a 1 to this bit location will mask IPL2 interrupts. Masking IPL2 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

BIT[16]: IM1-MASK IPL1  
 Writing a 1 to this bit location will mask IPL1 interrupts. Masking IPL1 will stop the interrupt that is normally sent to the 68020. The interrupt status can be tested by reading INTERRUPT CSR 2 register. Writing a 0 to this bit will allow this IPL interrupt to be sent to the 68020.

NOTE: IPL7 IS NON-MASKABLE

### 15.6.5 SOURCE INTERRUPT ENABLE REGISTER (SIER)

Adapter Manager ( 68020 ) Physical Address: \$0010 1408

Access: R/W ( Read / Write )

INITIALIZED AS: All bits zero.

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3   | 3   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 1   | 1   | 1   | 1   |     |
| 1   | 0   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   | 9   | 8   | 7   | 6   |
| NIO | NIO | NIO | NIO | NIO | RES | ESP | PMX | FCP | ELM | MAC | RMC | SMT | PAR | HWT | EVL |

Bit Descriptions:

- BIT[31:27]: NIO-NOT IMPLEMENTED, READ AS 0
- BIT[26]: RES-RESERVED  
Writing a 1 to this bit location will enable RESERVED interrupts. Disabling RESERVED will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[25]: ESP-ENABLE ESP CHIP INTERRUPT  
Writing a 1 to this bit location will enable ESP CHIP interrupts. Disabling ESP CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[24]: PMX-ENABLE PM START TRANSMIT INTERRUPT  
Writing a 1 to this bit location will enable PM START TRANSMIT interrupts. Disabling PM START TRANSMIT will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[23]: FCP-ENABLE FDDI CSR BUS PARITY ERROR INTERRUPT  
Writing a 1 to this bit location will enable FDDI CSR BUS PARITY ERROR interrupts. Disabling FDDI CSR BUS PARITY ERROR will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[22]: ELM-ENABLE FDDI ELM CHIP INTERRUPT  
Writing a 1 to this bit location will enable FDDI ELM CHIP interrupts. Disabling FDDI ELM CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[21]: MAC-ENABLE FDDI MAC CHIP INTERRUPT  
Writing a 1 to this bit location will enable FDDI MAC CHIP interrupts. Disabling FDDI MAC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

- BIT[20]:** RMC-ENABLE FDDI RMC CHIP INTERRUPT  
 Writing a 1 to this bit location will enable FDDI RMC CHIP interrupts. Disabling FDDI RMC CHIP will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[19]:** SMT-ENABLE SMT PRIORITY INTERRUPT  
 Writing a 1 to this bit location will enable SMT PRIORITY interrupts. Disabling SMT PRIORITY will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[18]:** PAR-ENABLE PARSER INTERRUPT  
 Writing a 1 to this bit location will enable PARSER interrupts. Disabling PARSER will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[17]:** HWT-ENABLE HARDWARE TIMER INTERRUPT  
 Writing a 1 to this bit location will enable HARDWARE TIMER interrupts. Disabling HARDWARE TIMER will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.
- BIT[16]:** EVL-ENABLE EVENT LEVEL MONITOR INTERRUPT  
 Writing a 1 to this bit location will enable EVENT LEVEL MONITOR interrupts. Disabling EVENT LEVEL MONITOR will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

**15.6.6 IPL INTERRUPT ENABLE REGISTER (IPLIER)**

Adapter Manager ( 68020 ) Physical Address: \$0010 140A

Access: R/W ( Read / Write )

INITIALIZED AS: All bits zero.

|                                                                                             |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|---------------------------------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 3                                                                                           | 3   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 2   | 1   | 1   | 1   | 1   |     |     |
| 1                                                                                           | 0   | 9   | 8   | 7   | 6   | 5   | 4   | 3   | 2   | 1   | 0   | 9   | 8   | 7   | 6   |     |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| NIO                                                                                         | NIO | NIO | NIO | NIO | NIO | NIO | NIO | NIO | NIO | NIO | IE6 | IE5 | IE4 | IE3 | IE2 | IE1 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

Bit Descriptions:

- BIT[31:22]:** NIO-NOT IMPLEMENTED, READ AS 0
- BIT[21]:** IE6-ENABLE IPL6 INTERRUPTS  
 Writing a 1 to this bit location will enable IPL6 interrupts. Disabling IPL6 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.



BIT[20]: IE5-ENABLE IPL5 INTERRUPTS  
Writing a 1 to this bit location will enable IPL5 interrupts. Disabling IPL5 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[19]: IE4-ENABLE IPL4 INTERRUPTS  
Writing a 1 to this bit location will enable IPL4 interrupts. Disabling IPL4 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[18]: IE3-ENABLE IPL3 INTERRUPTS  
Writing a 1 to this bit location will enable IPL3 interrupts. Disabling IPL3 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[17]: IE2-ENABLE IPL2 INTERRUPTS  
Writing a 1 to this bit location will enable IPL2 interrupts. Disabling IPL2 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

BIT[16]: IE1-ENABLE IPL1 INTERRUPTS  
Writing a 1 to this bit location will enable IPL1 interrupts. Disabling IPL1 will stop the interrupt that is normally sent to the 68020. The interrupt status is NOT indicated in the INTERRUPT CSR 1 register. There is no status that this interrupt occurred.

### 15.6.7 INTERRUPT SELF TEST REGISTER (INTST)

Adapter Manager ( 68020 ) Physical Address: \$0010 140C

Access: R/W ( Read / Write )

INITIALIZED AS: All bits zero.

```

  3  3  2  2  2  2  2  2  2  2  2  2  1  1  1  1
  1  0  9  8  7  6  5  4  3  2  1  0  9  8  7  6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|NIO|NIO|WDT|ACF|AEP|RES|ESP|PMX|FCP|ELM|MAC|RMC|SMT|PAR|HWT|EVL|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

BIT[31:30]: NIO-NOT IMPLEMENTED, READ AS 0

Digital Equipment Corporation - VAX Products and Options  
Confidential and Proprietary

- BIT[29]:       WDT-TEST WATCHDOG TIMER INTERRUPT  
Writing a 1 to this bit location will force a WATCHDOG TIMER interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[28]:       ACF-TEST AC POWER FAIL INTERRUPT  
Writing a 1 to this bit location will force a AC POWER FAIL interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[27]:       AEP-TEST AMI/ESP BUS PARITY ERROR INTERRUPT  
Writing a 1 to this bit location will force a AMI/ESP BUS PARITY ERROR interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[26]:       RES-RESERVED  
Writing a 1 to this bit location will force a RESERVED interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[25]:       ESP-TEST ESP CHIP INTERRUPT  
Writing a 1 to this bit location will force a ESP CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.

- BIT[24]: PMX-TEST PM START TRANSMIT INTERRUPT  
Writing a 1 to this bit location will force a PM START TRANSMIT interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[23]: FCP-TEST FDDI CSR BUS PARITY ERROR INTERRUPT  
Writing a 1 to this bit location will force a FDDI CSR BUS PARITY ERROR interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[22]: ELM-TEST FDDI ELM CHIP INTERRUPT  
Writing a 1 to this bit location will force a FDDI ELM CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[21]: MAC-TEST FDDI MAC CHIP INTERRUPT  
Writing a 1 to this bit location will force a FDDI MAC CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[20]: RMC-TEST FDDI RMC CHIP INTERRUPT  
Writing a 1 to this bit location will force a FDDI RMC CHIP interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.

- BIT[19]: SMT-TEST SMT PRIORITY INTERRUPT  
Writing a 1 to this bit location will force a SMT PRIORITY interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select of GPCSR must be set first.
- BIT[18]: PAR-TEST PARSER INTERRUPT  
Writing a 1 to this bit location will force a PARSER interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is a level sensitive interrupt. This bit is considered the source of the interrupt and must be cleared during the interrupt service routine. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[17]: HWT-TEST HARDWARE TIMER INTERRUPT  
Writing a 1 to this bit location will force a HARDWARE TIMER interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.
- BIT[16]: EVL-TEST EVENT LEVEL MONITOR INTERRUPT  
Writing a 1 to this bit location will force a EVENT LEVEL MONITOR interrupt. This bit is used by self-test to check the interrupt priority encoder logic in the AMI Gate Array. The interrupt status can be tested by reading the INTERRUPT CSR 1 register. This is an edge triggered interrupt; writing a 1 to this bit will cause a pulse that will generate an interrupt. The interrupt will be cleared during an IACK cycle. Reading this bit will return a 0. To use this bit to force interrupts bit 14 (Self-Test Select) of GPCSR must be set first.

## 15.6.8 VECTOR REGISTERS (VECREGS)

WATCHDOG TIMER VECTOR REGISTER (WTVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1500  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                       |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                     | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |   |
| 1                     | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |
| WATCHDOG TIMER VECTOR |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
=====

BIT[31:24]: WATCHDOG TIMER VECTOR  
This area is to be programmed with the vector that will be used for WATCHDOG TIMER interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

AC POWER FAIL VECTOR REGISTER (ACPFVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1502  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                    | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |   |
| 1                    | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |
| AC POWER FAIL VECTOR |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
=====

BIT[31:24]: AC POWER FAIL VECTOR  
This area is to be programmed with the vector that will be used for AC POWER FAIL interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

AMI/ESP BUS PARITY ERROR VECTOR REGISTER (ABPEVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1504  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                               | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |   |
| 1                               | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |
| AMI/ESP BUS PARITY ERROR VECTOR |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

**Bit Descriptions:**  
=====

BIT[31:24]: AMI/ESP BUS PARITY ERROR VECTOR  
This area is to be programmed with the vector  
that will be used for AMI/ESP BUS PARITY ERROR interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**ESP CHIP VECTOR REGISTER (ECVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1506  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|-----------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3               | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |   |
| 1               | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |   |
| ESP CHIP VECTOR |   |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit Descriptions:**  
=====

BIT[31:24]: ESP CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for ESP CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**PARSER VECTOR REGISTER (PVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1508  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3             | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |   |
| 1             | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |   |
| PARSER VECTOR |   |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Bit Descriptions:**  
=====

BIT[31:24]: PARSER VECTOR  
This area is to be programmed with the vector  
that will be used for PARSER interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**FDDI ELM CHIP VECTOR REGISTER (FECVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 150A  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.

=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI ELM CHIP VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

=====

BIT[31:24]: FDDI ELM CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI ELM CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

FDDI MAC CHIP VECTOR REGISTER (FMCVR)

=====

Adapter Manager ( 68020 ) Physical Address: \$0010 150C

=====

Access: R/W ( Read / Write )

=====

INITIALIZED AS: All bits zero.

=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI MAC CHIP VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

=====

BIT[31:24]: FDDI MAC CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI MAC CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

FDDI RMC CHIP VECTOR REGISTER (FRCVR)

=====

Adapter Manager ( 68020 ) Physical Address: \$0010 150E

=====

Access: R/W ( Read / Write )

=====

INITIALIZED AS: All bits zero.

=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI RMC CHIP VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:

=====

BIT[31:24]: FDDI RMC CHIP VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI RMC CHIP interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

**SMT PRIORITY VECTOR REGISTER (SMPVVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1510  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                     |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                   | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |
| 1                   | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |
| SMT PRIORITY VECTOR |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
=====

BIT[31:24]: SMT PRIORITY VECTOR  
This area is to be programmed with the vector  
that will be used for SMT PRIORITY interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**PM START TRANSMIT VECTOR REGISTER (PMSTVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1512  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

|                          |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|--------------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                        | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |
| 1                        | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |
| PM START TRANSMIT VECTOR |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
=====

BIT[31:24]: PM START TRANSMIT VECTOR  
This area is to be programmed with the vector  
that will be used for PM START TRANSMIT interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**HARDWARE TIMER VECTOR REGISTER (HTVR)**  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1514  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====



Digital Equipment Corporation - VAX Products and Options  
Confidential and Proprietary

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   HARDWARE TIMER VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT[31:24]: HARDWARE TIMER VECTOR  
This area is to be programmed with the vector  
that will be used for HARDWARE TIMER interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

EVENT LEVEL MONITOR VECTOR REGISTER (EVLMMVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1516  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| EVENT LEVEL MONITOR VECTOR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT[31:24]: EVENT LEVEL MONITOR VECTOR  
This area is to be programmed with the vector  
that will be used for EVENT LEVEL MONITOR interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

FDDI PARITY ERROR VECTOR REGISTER (FPEMVR)  
=====

Adapter Manager ( 68020 ) Physical Address: \$0010 1518  
=====

Access: R/W ( Read / Write )  
=====

INITIALIZED AS: All bits zero.  
=====

```

  3 3 2 2 2 2 2 2 2 2 2 1 1 1 1
  1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   FDDI PARITY ERROR VECTOR   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Bit Descriptions:  
=====

BIT[31:24]: FDDI PARITY ERROR VECTOR  
This area is to be programmed with the vector  
that will be used for FDDI PARITY ERROR interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

FDDI RESERVED VECTOR REGISTER (FRVR)  
 =====

Adapter Manager ( 68020 ) Physical Address: \$0010 151A  
 =====

Access: R/W ( Read / Write )  
 =====

INITIALIZED AS: All bits zero.  
 =====

|                      |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|----------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3                    | 3 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 |   |   |   |
| 1                    | 0 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 9 | 8 | 7 | 6 |   |   |   |
| FDDI RESERVED VECTOR |   |   |   |   |   |   |   |   |   |   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit Descriptions:  
 =====

BIT[31:24]: FDDI RESERVED VECTOR  
 This area is to be programmed with the vector  
 that will be used for FDDI RESERVED interrupt.

BIT[23:16]: NIO-NOT IMPLEMENTED, READ AS 0



**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

| ESPAD                                                                     |   |   |   |   | ESPD  |   |   |   |   |   |   |   |
|---------------------------------------------------------------------------|---|---|---|---|-------|---|---|---|---|---|---|---|
| =====                                                                     |   |   |   |   | ===== |   |   |   |   |   |   |   |
| 0                                                                         | 0 | 0 | 0 | 0 | 3     | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| 5                                                                         | 4 | 3 | 2 | 1 | 0     | 4 | 3 | 6 | 5 | 8 | 7 | 0 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 0 | 0 | 1     | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| LOW LONGWORD                                                              |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 0 | 0 | 1     | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| HIGH LONGWORD                                                             |   |   |   |   |       |   |   |   |   |   |   |   |

**ESP/XMI Data Registers**

| ESPAD                                                                     |   |   |   |   | ESPD  |   |   |   |   |   |   |   |
|---------------------------------------------------------------------------|---|---|---|---|-------|---|---|---|---|---|---|---|
| =====                                                                     |   |   |   |   | ===== |   |   |   |   |   |   |   |
| 0                                                                         | 0 | 0 | 0 | 0 | 3     | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| 5                                                                         | 4 | 3 | 2 | 1 | 0     | 4 | 3 | 6 | 5 | 8 | 7 | 0 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 0 | 0     | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| Longword of Data for XMI                                                  |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 0 | 0     | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| Longword of Data for XMI                                                  |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 0 | 1     | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| Longword of Data for XMI                                                  |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 0 | 1     | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| Longword of Data for XMI                                                  |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |

**ESP XMI/PORT Registers**

| ESPAD                                                                     |   |   |   |   | ESPD  |   |   |   |   |   |   |   |
|---------------------------------------------------------------------------|---|---|---|---|-------|---|---|---|---|---|---|---|
| =====                                                                     |   |   |   |   | ===== |   |   |   |   |   |   |   |
| 0                                                                         | 0 | 0 | 0 | 0 | 3     | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| 5                                                                         | 4 | 3 | 2 | 1 | 0     | 4 | 3 | 6 | 5 | 8 | 7 | 0 |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 1 | 0     | 0 | 1 | 1 | 0 | 0 | 0 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| XDEV                                                                      |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 1 | 0     | 0 | 1 | 1 | 0 | 0 | 0 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| XBER                                                                      |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 1 | 1     | 1 | 1 | 1 | 0 | 0 | 0 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| XFADR                                                                     |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 0 | 1 | 1 | 1     | 1 | 1 | 1 | 1 | 1 | 1 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| XFAER                                                                     |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| 1                                                                         | 0 | 1 | 0 | 0 | 0     | 0 | 0 | 0 | 0 | 0 | 0 |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |
| XPD1                                                                      |   |   |   |   |       |   |   |   |   |   |   |   |
| +-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+ |   |   |   |   |       |   |   |   |   |   |   |   |

| ESPAD       |   |     |  | ESPD  |  |     |  |   |  |
|-------------|---|-----|--|-------|--|-----|--|---|--|
| =====       |   |     |  | ===== |  |     |  |   |  |
| 0 0 0 0 0 0 | 3 | 2 2 |  | 1 1   |  | 0 0 |  | 0 |  |
| 5 4 3 2 1 0 | 1 | 4 3 |  | 6 5   |  | 8 7 |  | 0 |  |
| 0 1 0 0 0 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 0 0 1 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 0 0 1 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 0 1 0 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 0 1 0 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 0 1 1 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 0 1 1 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 0 0 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 0 0 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 0 1 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 0 1 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 1 0 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 1 0 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 1 1 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 0 1 1 1 1 1 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
|             |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |
| 1 0 0 0 0 0 |   |     |  |       |  |     |  |   |  |
| -----       |   |     |  |       |  |     |  |   |  |



**Digital Equipment Corporation - VAX Products and Options  
Confidential and Proprietary**

BIT<31:00>: Bit Settings dependent on ESP/XMI Operation  
Refer to ESP Spec for details.

ESP/XMI Data Registers

| 68020<br>Address | 3<br>1 | 1 1<br>6 5                   | 0<br>0 |
|------------------|--------|------------------------------|--------|
| \$0010 0820      |        | Longword of Data for ESP/XMI |        |
| \$0010 0824      |        | Longword of Data for ESP/XMI |        |
| \$0010 0828      |        | Longword of Data for ESP/XMI |        |
| \$0010 082C      |        | Longword of Data for ESP/XMI |        |

XMI/PORT Registers - as seen by 68020.

| 68020<br>Address | 3<br>1 | 1 1<br>6 5     | 0<br>0 |
|------------------|--------|----------------|--------|
| \$0010 0830      |        | XDEV Register  |        |
| \$0010 0834      |        | XBER Register  |        |
| \$0010 0838      |        | XFADR Register |        |
| \$0010 083C      |        | XFAER Register |        |
| \$0010 0840      |        | XPD1 Register  |        |
| \$0010 0844      |        | XPD2 Register  |        |
| \$0010 0848      |        | XPST Register  |        |
| \$0010 084C      |        | XPUD Register  |        |
| \$0010 0850      |        | TRANS_SLEEP_LO |        |
| \$0010 0854      |        | TRANS_SLEEP_HI |        |

**Digital Equipment Corporation - VAX Products and Options**  
**Confidential and Proprietary**

|             |                                                           |
|-------------|-----------------------------------------------------------|
| \$0010 0858 | REV_SLEEP_LO                                              |
| \$0010 085C | REV_SLEEP_HI                                              |
| \$0010 0860 | INTR_VECT                                                 |
| \$0010 0864 | IDENT_VECT                                                |
| \$0010 0868 | 68_INTR                                                   |
| \$0010 086C | ESP_MISC                                                  |
| \$0010 0870 | Host Transmit Ring Base Low Address                       |
| \$0010 0874 | Host Transmit Ring Base High Addr, Ring Size & Entry Size |
| \$0010 0878 | Host Receive Ring Base Low Address                        |
| \$0010 087C | Host Receive Ring Base High Addr, Ring Size & Entry Size  |
| \$0010 0880 | PMC Transmit Ring Base Address and Ring Size              |
| \$0010 0884 | PMC Receive Ring Base Address and Ring Size               |
| \$0010 0888 | ESP Error                                                 |



## Updating the DEC FDDIcontroller 400

This manual explains how to install the software microcode onto your VMS load host using VMSINSTAL and how to update the DEC FDDIcontroller 400 (DEMFA) using the Digital Network Device Upgrade (DECndu) utility.

### NOTE

To install the software microcode, you must refer to the release notes for the current version number of the firmware. DEMFA $nnn$  is used throughout the installation procedure. Use the current version number in place of  $nnn$ .

This manual assumes that you have installed the current version of DECndu onto the load host.

### Intended Audience

This manual is intended for the system or network manager responsible for updating network devices on a VAX system running the VMS operating system.

### Documentation

The update documentation contains the following:

- The *DEC FDDIcontroller 400 Software Microcode Installation on VMS* manual (this manual)
- Release notes (also available on-line in ASCII format)

## Associated Documentation

The following documentation contains additional information you may need to install the software microcode.

- *DEC FDDIcontroller 400 Installation manual* (P/N EK-DEMFA-IN)

This manual explains how to install and troubleshoot the DEC FDDIcontroller 400.

- *Digital Network Device Upgrade Utility Installation on VMS manual* (P/N AA-PEJAC-TE)

This manual tells you how to install the DECndu utility onto the VMS load host using VMSINSTAL.

- DECndu release notes

## Structure of This Manual

This manual contains six sections, as follows:

- ① Lists the steps to take before installing the software microcode. page 3
- ② Explains how to install the software microcode onto the VMS load host using VMSINSTAL. page 3
- ③ Lists the steps to take before updating the DECconcentrator 500. page 6
- ④ Explains how to update the DEC FDDIcontroller 400 using the DECndu utility. page 7
- ⑤ Explains how to verify the update of the DEC FDDIcontroller 400. page 8
- ⑥ Lists the DEC FDDIcontroller 400 software image files you receive with the update. page 9

## Installation Time

The installation time is 5-10 minutes.

## ① Preparing to Install The Software Microcode

Before installing the software microcode, do the following:

1. Read the DECndu and DEC FDDIcontroller 400 release notes shipped with the update kit.
2. Record the DECndu version number and the DEC FDDIcontroller 400 version number.
3. Ensure that you are running VMS V5.3<sup>-4</sup> or higher.
4. Ensure that you have the VMS Required Saveset and the Network Support Saveset classes installed on the system.
5. Determine which systems are designated as load hosts.
6. Verify that there is enough free disk space on the load host. Refer to Table 1.
7. Back up the system disk.

### NOTE

The update procedure updates the adapters locally (not over the network).

Table 1: Disk Space

| Software                                                 | Blocks Peak |
|----------------------------------------------------------|-------------|
| DEC FDDIcontroller software image files on the load host | 2000        |

## ② Installing the Software Microcode

The software microcode update consists of installing the software onto the load host and then using DECndu to update the DEC FDDIcontroller 400.

To get help with a question, type a question mark (?) and press .

**NOTE**

The installation procedures use *nnn* for the version numbers of DECndu and the DEC FDDIcontroller 400. Refer to the release notes for the current version numbers.

**NOTE**

The examples in this manual use *MUA0* as your load device. Substitute the actual load device as appropriate to your system configuration.

Install the DEC FDDIcontroller 400 software onto the load host as follows:

1. Determine if DECndu supports the device by typing the following:

```
$ Download list
```

**NOTE**

You can not update a device that is not on the list. Contact your Digital representative for more information.

2. Insert the "DEC FDDIcontroller 400" software distribution media into the appropriate load device. Ensure that the drive is on-line.
3. Start the installation procedure by typing the following:

```
$ @SYS$UPDATE:VMSINSTAL DEMFAnnn MUA0: 
```

4. Respond to the VMSINSTAL prompts. Default responses are in brackets. Example 1 shows you the installation procedure.

## Example 1: Installing DEC FDDIcontroller 400 Software Image Files

```
$ @SYS$UPDATE:VMSINSTAL DEMFAnnn MUA0: 
```

VAX/VMS Software Product Installation Procedure  
V5.3-2

It is 24-Feb-92 at 10:55.

Enter a question mark (?) at any time for help.

\* Are you satisfied with the backup of your  
system disk [YES] ?

The following products will be processed:

DEMFA Vn.n

Beginning installation of DEMFA Vn.n at 10:55

%VMSINSTAL-I-RESTORE, Restoring product saveset  
A...

%VMSINSTAL-I-REMOVED, The product's release  
notes have been successfully moved to SYS\$HELP.  
DEMFA Vn.n Installation is commencing...

\* Do you want to run the IVP after the instal-  
lation [YES]?

\* Do you want to purge files replaced by this  
installation [YES]?

All the questions regarding the installation  
have been asked. The installation will contin-  
ue for about another 2-4 minutes.

DEMFA Vn.n Installation completed.

The DEMFA Vn.n kit includes:

```
MOM$LOAD:DEMFAnnn.SYS          [new]
SYS$HELP:DEMFAnnn.RELEASE_NOTES [new]
SYS$TEST:DEMFA$IVP.COM         [new]
```

Please be sure to read the product release  
notes and the software installation manual for  
proper installation of this product.

The installation verification procedure for  
this product is DEMFA\$IVP.COM and is located in  
the SYS\$TEST directory. It may be executed by  
typing the following command:

```
$ @SYS$TEST:DEMFA$IVP.COM
```

%VMSINSTAL-I-MOVEFILES, files will now be moved  
to their target directories...

DEMFA Vn.n. Installation Verification Procedure  
commencing...

Successfully located DEMFAnnn.SYS  
Successfully located DEMFAnnn.RELEASE\_NOTES in  
SYS\$HELP  
DEMFA Vn.n Installation Verification Procedure  
completed successfully.

Installation of DEMFA Vn.n completed at 10:58  
VMSINSTAL procedure done at 10:59  
\$

---

### **3** Preparing to Update the DEC FDDIcontroller 400

Before performing the update procedure, read the following:

- Verify and record the DEC FDDIcontroller 400 hardware address.
- You must reset the adapter to activate the software after the update. There are two methods to activate the software, as follows:
  - Method 1  
Issue the /RESET command qualifier in the command line to update the software (default is /NO-RESET). Note that the DEC FDDIcontroller shuts down for approximately 2 minutes.
  - Method 2  
Perform a system shutdown and restart after the software update is complete.

#### **NOTE**

The /RESET command qualifier terminates all network links on reset. Digital suggests that you perform the update during a planned system shutdown.

- If you are using a VAX 6000, ensure that the system control panel switch is in the update position.
- If you are using a VAX 9000, ensure that you are in console mode. Set the XMI update switch to ON, by typing the following:

```
$ SET XMI_UPDATE ON 
```

## 4 Using DECndu to Update the DEC FDDIcontroller 400

The DECndu update procedure allows you to choose command qualifiers for specific information. Table 2 lists the DECndu command qualifiers used in the following examples.

For help information on DECndu, type the following:

```
$ HELP DOWNLOAD 
```

**Table 2: DECndu Command Qualifiers**

| Qualifier/Keyword | Use                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DOWNLOAD LOAD     | Starts the update procedure.                                                                                                                                                     |
| /CONFIRM          | DECndu prompts you to continue after receiving information about the device.                                                                                                     |
| /RESET            | Resets the DEC FDDIcontroller 400 and activates the software. This qualifier shuts down the DEC FDDIcontroller 400 for approximately 2 minutes and terminates all network links. |
| /NORESET          | The DEC FDDIcontroller 400 is not reset and the software is not activated. You must perform a system shutdown and restart after the software update is complete.                 |

### NOTE

The example uses *FXA0* for the adapter name.

To update the DEC FDDIcontroller 400, perform the following three steps:

1. Start the update procedure by performing the following three steps:

```
$ DOWNLOAD LOAD/CONFIRM/RESET FXA0  
MOM$LOAD:DEMFAnnn.SYS
```

2. Press .

3. Set the XMI update switch to off, type the following:

```
$ SET XMI_UPDATE OFF 
```

The update begins. Example 2 shows the screen display when a full command line is used to update the DEC FDDIcontroller 400. Note that your text will differ if you do not use a full command line.

### Example 2: Updating the DEC FDDIcontroller 400

---

```
$ DOWNLOAD LOAD/CONFIRM/RESET FXA0 MOM$LOAD:DEM  
FAnnn.SYS 
```

```
(C) Digital Equipment Corporation. 1992. All  
Rights Reserved.
```

```
%NDU-I-TARGET_VERSION, Target device is a DEMFA  
version n.n
```

```
Do you want to continue? (y/n) y 
```

```
%NDU-I-TARGET_VERSION, Target device is a DEMFA  
version n.n
```

```
$
```

---

### Verifying the Update

To verify the device and firmware version, type the following:

```
$ DOWNLOAD SHOW DEVICE FXA0 
```

Example 3 shows the update verification display.

### Example 3: Verifying the Update

---

```
$ DOWNLOAD SHOW DEVICE FXA0 
```

```
(C) Digital Equipment Corporation. 1992. All  
Rights Reserved.
```

```
Name          DEMFA  
Firmware Rev          Vn.n  
$
```

---



## DEC FDDIcontroller 400 Software Image Files

Table 3 lists the software image files that you receive.

**Table 3: DEC FDDIcontroller 400 Software Image Files**

| <b>File Name</b>                    | <b>Description</b>                                              |
|-------------------------------------|-----------------------------------------------------------------|
| DEMFA <sub>nnn</sub> .SYS           | DEC FDDIcontroller 400 image file in the MOM\$LOAD directory    |
| DEMFA\$IVP.COM                      | Installation Verification Procedure in the SYS\$TEST directory  |
| DEMFA <sub>nnn</sub> .RELEASE_NOTES | DEC FDDIcontroller 400 release notes in the SYS\$HELP directory |

## Updating the DEC FDDIcontroller 700

This manual explains how to install the software microcode onto your ULTRIX load host and how to update the DEC FDDIcontroller 700 (defza) using the Digital Network Device Upgrade (DECndu) utility.

### NOTE

To install the software microcode, refer to the release notes for the device name and current version number of the firmware. *defzannn* is used throughout the installation procedure.

This manual assumes that you have installed the current version of DECndu onto the load host.

### Intended Audience

This manual is intended for the system or network manager responsible for updating network devices on an ULTRIX RISC operating system.

### Documentation

The update documentation contains the following:

- The *DEC FDDIcontroller 700 Software Microcode Installation on ULTRIX* manual (this manual)
- Release notes (also available on-line in ASCII format)

## Associated Documentation

The following documentation contains additional information you may need to install the software microcode.

- *Digital Network Device Upgrade Utility Installation on ULTRIX* manual (P/N AA-PELHC-TE)

This manual tells you how to install the DECndu utility onto the ULTRIX load host.

- *DEC FDDIcontroller 700 Installation* manual (P/N EK-DEFZA-IN)

This manual explains how to install the DEC FDDIcontroller 700 and also includes information about the switches.

- DECndu release notes

## Structure of This Manual

This manual contains six sections, as follows:

- ① Lists the steps to take before installing the software microcode. page 3
- ② Explains how to install the software microcode onto the load host. page 4
- ③ Explains how to verify the software microcode installation. page 5
- ④ Explains how to update the DEC FDDIcontroller 700. page 6
- ⑤ Explains how to verify the update of the DEC FDDIcontroller 700. page 7
- ⑥ Lists the DEC FDDIcontroller 700 software image files you receive with the update. page 8

## Installation Time

The installation time is 5-10 minutes.



## Preparing to Install the Software Microcode

Before you installing the software microcode, do the following:

1. Read the DECndu and DEC FDDIcontroller 700 release notes shipped with the update kit.
2. Record the DECndu version number and the DEC FDDIcontroller 700 version number.
3. Ensure that you are running ULTRIX V4.0 or higher.
4. Ensure that you have UDTBASE 4xx installed on the system.
5. Determine which systems are designated as load hosts.
6. Verify that the load host has enough free disk space. Refer to Table 1.
7. Back up the system disk.

### NOTE

The update procedure only update the DEC FDDIcontroller 700 locally (not over the network).

Table 1: Disk Space on ULTRIX RISC

| Subset Name | /File System | /user File System | /var File System |
|-------------|--------------|-------------------|------------------|
| FZAMCnnn    | 0 KB         | 310 KB            | 0 KB             |

## ② Installing the Software Microcode

The software microcode update consists of installing the software onto the ULTRIX load host and then using DECndu to update the DEC FDDIcontroller 700.

For help on DECndu, type `man decndu` at the prompt and press `Return`.

Install the DEC FDDIcontroller 700 software image files onto the load host, as follows:

1. Determine whether DECndu supports the device by typing the following:

```
# decndu -l
```

### NOTE

**You cannot update a device that is not on the list. contact your Digital representative for more information.**

2. Insert the "DEC FDDIcontroller 700" software distribution media labeled into the load device.
3. Start the installation, type the following:

```
# setld -l /dev/rnmt0h Return
```

4. Respond to the prompts. Refer to Example 1.

### Example 1: Installing the Software Microcode

---

```
# setld -l /dev/rnmt0h Return
Please make sure your installation tape is
mounted and on-line.
Are you ready (y/n) y Return
Positioning Tape
***Enter Subset Selections***
The subsets listed below are optional:
1) DEC FDDIcontroller 700
2) All of the above
3) None of the above
4) Exit without installing subsets
Enter your choice(s): 1 Return
You are installing the following subsets:
  DEC FDDIcontroller 700
Is this correct? (y/n) y Return

Beginning installation of DEC FDDIcontroller
700 Vn.n
Copying DEC FDDIcontroller 700 (FZAMCnnn) from
tape
```

```
Verifying DEC FDDIcontroller 700 (FZAMCnnn)
Successful installation of DEC FDDIcontroller
700 Vn.n
Would you like to run the IVP after the instal-
lation? (Y/N) y 
```

```
Beginning DEC FDDIcontroller 700 Vn.n IVP
Successfully found ./usr/kits/FZAnnn/def-
zannn.release_notes
Successfully found ./usr/kits/FZAnnn/def-
zannn.sys
(c) Digital Equipment Corporation. 1992. All
Rights Reserved.
Name          DEFZA
Firmware Rev  Vn.n
Successful completion of DEC FDDIcontroller 700
Vn.n
Rewinding Tape....
Working....Wed march 13 10:51:20 EST 1992
#
```

---

### ④ Verifying the Installation

To verify the software microcode installation, type the following:

```
# setld -v FZAMCnnn 
```

Example 2 shows the installation verification display.

#### Example 2: Installation Verification

---

```
# setld -v FZAMCnnn 

Beginning DEC FDDIcontroller 700 Vn.n IVP

Successfully found ./usr/kits/FZAnnn/def-
zannn.release_notes
Successfully found ./usr/kits/FZAnnn/def-
zannn.sys
(c) Digital Equipment Corporation. 1992. All
Rights Reserved.
Firmware Rev          DEFZA

Successful completion of DEC FDDIcontroller
700 Vn.n IVP
#
```

---

## 4 Using DECndu to Update the DEC FDDIcontroller 700

The DECndu update procedure allows you to choose command qualifiers for specific information. Table 2 lists the DECndu command qualifiers used in the following examples.

For help information on DECndu, type the following:

```
# man decndu
```

**Table 2: DECndu Command Qualifiers**

| Qualifier | Meaning                                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------|
| -i        | Displays informational messages during the update.                                                                                         |
| -s        | Shows device information.                                                                                                                  |
| -u        | Updates the device with the specified image. The -u qualifier must be followed by the hardware address, for example, -u 08-00-2b-11-22-33. |
| -v        | If you want DECndu to prompt you to continue after receiving information about the device.                                                 |
| -V        | Displays the version of the DECndu utility.                                                                                                |

### NOTE

The examples use *fza0* for the adapter name.

To update the DEC FDDIcontroller 700, include the full pathname (*/usr/kits/FZAnnn*) and image file name (*defzannn.sys*):

1. Start the update procedure; type:

```
# decndu -i -v -u fza0 ./usr/kits/FZAnnn/defzannn.sys
```

2. Press **Return**

Example 3 shows the screen display for the update.

### Example 3: Updating the DEC FDDIcontroller 700 on ULTRIX RISC

---

```
# decndu -i -v -u fza0 /usr/kits/FZAann/def-  
za.sys 
```

(c) Digital Equipment Corporation. 1992. All  
Rights Reserved.

DECndu: Target is a DEFZA version n.n

Do you want to continue? (y/n) y

DECndu: Target is a DEFZA version n.n

```
#
```

---



### Verifying the Update of the DEC FDDIcontroller 700

If you did not use the -i command qualifier and want to  
verify the device and firmware version, enter the -s com-  
mand and the name of the adapter as follows:

```
# decndu -s fza0
```

Example 4 shows the update verification screen display.

### Example 4: Verifying the Update

---

```
# decndu -s fza0 
```

(c) Digital Equipment Corporation. 1992. All  
Rights Reserved.

|              |       |
|--------------|-------|
| Name         | DEFZA |
| Firmware Rev | Vn.n  |

---





## DEC FDDIcontroller 700 Software Image Files

Table 3 lists the software image files that you receive.

**Table 3: DEC FDDIcontroller 700 Software Image  
Files on an ULTRIX RISC System**

| <b>File Name</b>                    | <b>Description</b>                                                                                                                                    |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>defzannn.sys</code>           | Software image file for the device in the <code>/usr/kits/FZA<math>n</math>n</code> directory that ULTRIX uses for part of the installation procedure |
| <code>defzannn.release_notes</code> | Release notes for <code>V<math>n</math>.<math>n</math></code> in <code>/usr/lits/FZA<math>n</math>n</code> directory                                  |

TOP VIEW, PINS DOWN

|    |                    |                    |                    |                  |                     |                   |                   |                   |                   |                   |                    |                    |                    |                |
|----|--------------------|--------------------|--------------------|------------------|---------------------|-------------------|-------------------|-------------------|-------------------|-------------------|--------------------|--------------------|--------------------|----------------|
| 13 | A13 (90)<br>VDD    | B13 (87)<br>NPA1   | C13 (86)<br>NPA0   | D13 (83)<br>VSS  | E15 (81)<br>NPD15   | F13 (78)<br>NPD12 | G13 (77)<br>NPD11 | H13 (74)<br>NPD10 | J13 (71)<br>VDD   | K13 (69)<br>NPD6  | L13 (66)<br>NPD4   | M13 (63)<br>NPD1   | N13 (60)<br>VDD    |                |
| 12 | A12 (93)<br>NPA5   | B12 (92)<br>NPA4   | C12 (89)<br>NPA3   | D12 (85)<br>NPA2 | E12 (82)<br>VDD     | F12 (79)<br>NPD13 | G12 (75)<br>VSS   | H12 (73)<br>NPD9  | J12 (70)<br>NPD7  | K12 (67)<br>VSS   | L12 (65)<br>NPD3   | M12 (62)<br>NPD0   | N12 (57)<br>TXCTL1 |                |
| 11 | A11 (96)<br>VSS    | B11 (95)<br>TXDAT1 | C11 (91)<br>VDD    | D11 (88)<br>NPA2 | E11 (84)<br>MACESEL | F11 (80)<br>NPD14 | G11 (76)<br>VSS   | H11 (72)<br>NPD8  | J11 (68)<br>NPD5  | K11 (64)<br>NPD2  | L11 (61)<br>VDD    | M11 (59)<br>PWRUP  | N11 (56)<br>VSS    |                |
| 10 | A10 (99)<br>TXDAT4 | B10 (97)<br>TXDAT2 | C10 (94)<br>TXDAT0 |                  |                     |                   |                   |                   |                   |                   | L10 (58)<br>TXCTL6 | M10 (55)<br>TABORT | N10 (53)<br>NPEE   |                |
| 9  | A9 (101)<br>VDD    | B9 (100)<br>VDD    | C9 (98)<br>TXDAT3  |                  |                     |                   |                   |                   |                   |                   | L9 (54)<br>TARBY   | M9 (52)<br>TFRITY  | N9 (51)<br>TPATE0  |                |
| 8  | A8 (104)<br>VDD    | B8 (103)<br>VDD    | C8 (102)<br>TXDAT5 |                  |                     |                   |                   |                   |                   |                   | L8 (50)<br>VDD     | M8 (49)<br>TPATE1  | N8 (48)<br>TPATE2  |                |
| 7  | A7 (107)<br>VSS    | B7 (105)<br>VSS    | C7 (106)<br>VSS    |                  |                     |                   |                   |                   |                   |                   | L7 (46)<br>VSS     | M7 (45)<br>VSS     | N7 (47)<br>TPATE3  |                |
| 6  | A6 (108)<br>RDATA9 | B6 (109)<br>RDATA6 | C6 (110)<br>RDATA7 |                  |                     |                   |                   |                   |                   |                   | L6 (42)<br>RDATA6  | M6 (43)<br>RDATA5  | N6 (44)<br>RDATA4  |                |
| 5  | A5 (111)<br>VDD    | B5 (112)<br>VSS    | C5 (114)<br>RDATA5 |                  |                     |                   |                   |                   |                   |                   | L5 (38)<br>RDATA6  | M5 (40)<br>RDATA7  | N5 (41)<br>RDATA7  |                |
| 4  | A4 (113)<br>RDATA6 | B4 (115)<br>RDATA4 | C4 (118)<br>RDATA1 | D4<br>KEY        |                     |                   |                   |                   |                   |                   |                    | L4 (34)<br>RDATA3  | M4 (37)<br>RDATA5  | N4 (39)<br>VDD |
| 3  | A3 (116)<br>RDATA3 | B3 (119)<br>VDD    | C3 (1)<br>VDD      | D3 (4)<br>VSS    | E3 (8)<br>VSS       | F3 (12)<br>VSS    | G3 (16)<br>VDD    | H3 (20)<br>VDD    | J3 (24)<br>MTEST1 | K3 (26)<br>RDATA0 | L3 (31)<br>VDD     | M3 (35)<br>RDATA4  | N3 (36)<br>VSS     |                |
| 2  | A2 (117)<br>VSS    | B2 (2)<br>VSS      | C2 (5)<br>MTCOE    | D2 (7)<br>SYNCLK | E2 (10)<br>NFCLE    | F2 (13)<br>VDD    | G2 (15)<br>VSS    | H2 (19)<br>RCTL3  | J2 (22)<br>RCTL4  | K2 (25)<br>MTEST0 | L2 (29)<br>RDATA1  | M2 (32)<br>RDATA2  | N2 (33)<br>MTEST0  |                |
| 1  | A1 (120)<br>VDD    | B1 (3)<br>NPD16    | C1 (6)<br>LEADER   | D1 (9)<br>VDD    | E1 (11)<br>BYCLK    | F1 (14)<br>RCTL0  | G1 (17)<br>RCTL1  | H1 (18)<br>RCTL2  | J1 (21)<br>MSENC  | K1 (23)<br>RFRITY | L1 (26)<br>VSS     | M1 (27)<br>MSENC   | N1 (30)<br>VDD     |                |

DC7109B

Wed Oct 11 14:11:27 1989

INPUT
  OUTPUT
  BIDIR

TOSHIBA, 120PGA, T7075 (32K)

**digital**

COPY

## PMC PASS 2 Design Specification

### Revision X21 - The final draft version for PMC\_B

This document specifies the design of the second pass PMC Gate Array, being the Packet Memory Controller of the XFA (FDDI to XMI) Adapter. It is based on the original PMC design specification.

### Written by:

Ron Edgar / Joan Klebes

VAX PRODUCTS AND OPTIONS

February 1991

Digital Equipment Corporation

# CONTENTS

|                                                                  |           |
|------------------------------------------------------------------|-----------|
| Preface .....                                                    | v         |
| <b>Chapter 1 EXECUTIVE SUMMARY .....</b>                         | <b>1</b>  |
| 1.1 OVERVIEW .....                                               | 1         |
| 1.2 INTERFACES .....                                             | 3         |
| 1.2.1 RMC Interface .....                                        | 3         |
| 1.2.2 ESP Interface .....                                        | 3         |
| 1.3 FUNCTIONAL BLOCK DESCRIPTIONS .....                          | 3         |
| 1.3.1 PACKET BUFFER MEMORY .....                                 | 3         |
| 1.3.2 PAGE TABLE AND BUFFER DESCRIPTOR MEMORY .....              | 3         |
| 1.3.3 RING ENTRY MOVER .....                                     | 3         |
| 1.3.4 PMT - Packet Memory Test .....                             | 4         |
| 1.3.5 MIF - Memory Controller .....                              | 5         |
| 1.3.6 RMC I/F .....                                              | 5         |
| 1.3.7 ESP I/F .....                                              | 5         |
| 1.4 ADDITIONAL FEATURES .....                                    | 5         |
| 1.5 BYTE SWAPPING in the PMC .....                               | 6         |
| <b>Chapter 2 PMC CONTEXT .....</b>                               | <b>7</b>  |
| <b>Chapter 3 MEMORY INTERFACE AND PACKET BUFFER MEMORY .....</b> | <b>11</b> |
| 3.1 BANDWIDTH .....                                              | 11        |
| 3.2 DESIGN ATTRIBUTES .....                                      | 11        |
| 3.3 Minimum Sized Forwarded Packets .....                        | 11        |
| 3.4 CYCLE COUNTS .....                                           | 12        |
| 3.5 MIF REQUIRED INSTRUCTIONS .....                              | 15        |
| 3.6 MIF CONTEXT .....                                            | 16        |
| 3.7 MIF TIMING .....                                             | 19        |
| 3.8 EXPANSION OF MIF FUNCTIONALITY .....                         | 27        |
| 3.8.1 MEMORY CONTROL STATE MACHINE .....                         | 28        |
| 3.8.2 MIF ADDRESS MUX .....                                      | 28        |
| 3.8.3 MIF DMUX .....                                             | 28        |
| 3.8.4 MIF RAS/CAS MUX .....                                      | 28        |
| 3.8.5 CAS COUNTER .....                                          | 28        |
| 3.8.6 Misc MCSM bits .....                                       | 29        |
| 3.8.7 SRAM CALCULATE ADDRESS functional block .....              | 29        |
| 3.9 THE REFRESH FUNCTIONAL BLOCK .....                           | 29        |

|                                                                          |           |
|--------------------------------------------------------------------------|-----------|
| <b>Chapter 4 THE RING ENTRY MOVER</b> .....                              | <b>31</b> |
| <b>Chapter 5 RING ENTRY MOVER</b> .....                                  | <b>33</b> |
| 5.1 OVERVIEW .....                                                       | 33        |
| 5.2 THE PAGE TABLE .....                                                 | 33        |
| 5.2.1 LEGAL PATHS FOR PTEs BETWEEN PAGE TABLE RINGS .....                | 33        |
| 5.2.2 FILL AND FREE SPACE AT INITIALIZATION .....                        | 35        |
| 5.2.2.1 RING POINTERS .....                                              | 37        |
| 5.2.3 PAGE TABLE ENTRIES .....                                           | 37        |
| 5.2.3.1 PTE STRUCTURE AT INITIALIZATION .....                            | 38        |
| 5.3 SETTING THE STAGE FOR REM ACTIVITY .....                             | 39        |
| 5.4 BUFFER MANAGEMENT IN THE PACKET MEMORY CONTROLLER .....              | 40        |
| 5.4.1 Buffer Management for the HOST Interface .....                     | 40        |
| 5.4.2 Buffer Management for the ADAPTER MANAGER Interface .....          | 40        |
| 5.4.3 Buffer Management for the RMC Interface .....                      | 40        |
| 5.4.3.1 Buffer Allocation and RMC Packets .....                          | 41        |
| 5.4.3.2 Discarding a Packet .....                                        | 42        |
| 5.4.3.3 Buffer Allocation and Ring Configuration .....                   | 44        |
| 5.4.3.3.1 NOTE ON HOST BUFFER COUNT .....                                | 44        |
| 5.5 NOTIFYING THE INTERFACE AND WAKING THE MOVE_FREE STATE MACHINE ..... | 44        |
| 5.6 RING CONFIGURATIONS .....                                            | 45        |
| 5.7 REM FUNCTIONAL BLOCKS .....                                          | 46        |
| 5.7.1 EXPANSION OF THE REM .....                                         | 48        |
| 5.8 EXPANSION OF THE RING ENTRY MOVER .....                              | 49        |
| 5.8.1 MOVE_FILL BLOCK .....                                              | 49        |
| 5.8.1.1 MOVE_FILL SUPERVISORY STATE MACHINE FLOW .....                   | 49        |
| 5.8.1.2 OVERVIEW OF THE MOVE_FILL STATE MACHINE .....                    | 51        |
| 5.8.2 MOVE_FREE BLOCK .....                                              | 55        |
| 5.8.2.1 MOVE_FREE SUPERVISORY STATE MACHINE FLOW .....                   | 55        |
| 5.8.2.2 OVERVIEW OF THE MOVE_FREE STATE MACHINE .....                    | 57        |
| 5.8.3 ARBITER LOGIC BLOCK .....                                          | 61        |
| 5.9 PACKET TRACKING .....                                                | 63        |
| 5.10 REM FREEZE AND REM DISABLE .....                                    | 65        |
| 5.11 TWO EXCEPTION CASES .....                                           | 66        |
| 5.12 ADDITONAL REM FEATURES .....                                        | 67        |
| <b>Chapter 6 THE RMC INTERFACE</b> .....                                 | <b>69</b> |
| 6.1 RMC INTERFACE INSTRUCTION PROCESSING .....                           | 69        |
| 6.2 SPECIAL FEATURES OF THE RMC INTERFACE .....                          | 70        |

|                                                               |    |
|---------------------------------------------------------------|----|
| <b>Chapter 7 THE ESP BLOCKS</b> .....                         | 73 |
| 7.1 ESP BLOCK EXPANSION .....                                 | 73 |
| 7.1.1 THE SYNCHRONIZER .....                                  | 74 |
| 7.1.2 THE ESP INTERFACE FUNCTIONAL BLOCK .....                | 76 |
| 7.1.2.1 SPECIAL CYCLES .....                                  | 77 |
| 7.1.3 THE CSR FUNCTIONAL BLOCK .....                          | 77 |
| <b>Chapter 8 PMC REGISTERS</b> .....                          | 81 |
| 8.1 PMC REGISTER SUMMARY .....                                | 81 |
| 8.2 PMC REGISTER DEFINITIONS .....                            | 87 |
| 8.2.1 PMC CONTROL AND STATUS REGISTER (CSR00) .....           | 87 |
| 8.2.2 PMC RAM PARITY ERROR REGISTER (CSR01) .....             | 89 |
| 8.2.3 PMC MASK AND INTERRUPT REGISTER (CSR02) .....           | 90 |
| 8.2.4 PMC PROTOCOL ERROR REGISTER (CSR03) .....               | 91 |
| 8.2.5 PMC PMT FAILING ADDRESS REGISTER (CSR07) .....          | 93 |
| 8.2.6 PMC PMT FAILING DATA REGISTER (CSR0B) .....             | 94 |
| 8.2.7 PMC PMT CONTROL AND STATUS REGISTER (CSR0F) .....       | 94 |
| <b>Chapter 9 THE PACKET MEMORY (DRAM) TEST</b> .....          | 97 |
| 9.1 PACKET MEMORY TEST .....                                  | 97 |
| <b>Appendix A RING CONFIGURATIONS</b> .....                   | 99 |
| <b>FIGURES</b>                                                |    |
| 1-1 PACKET MEMORY CONTROLLER - FUNCTIONAL BLOCK DIAGRAM ..... | 2  |
| 1-2 MEMORY ALLOCATION .....                                   | 4  |
| 2-1 PMC CONTEXT .....                                         | 7  |
| 3-1 MINIMUM LLC PACKET CYCLE COUNT .....                      | 12 |
| 3-2 MIF SUPPORTED CYCLE TYPES .....                           | 15 |
| 3-3 MIF CONTEXT .....                                         | 16 |
| 3-4 DRAM READ BURST .....                                     | 19 |
| 3-5 DRAM WRITE BURST .....                                    | 20 |
| 3-6 READ BUFFER DESCRIPTOR .....                              | 21 |
| 3-7 WRITE BUFFER DESCRIPTOR .....                             | 22 |
| 3-8 READ PAGE TABLE ENTRY .....                               | 23 |
| 3-9 WRITE PAGE TABLE ENTRY (and Clear Own) .....              | 24 |
| 3-10 WRITE FORWARDING VECTOR, CLEAR OWN .....                 | 25 |
| 3-11 REFRESH .....                                            | 26 |
| 3-12 MIF FUNCTIONAL BLOCK DIAGRAM .....                       | 27 |
| 3-13 REFRESH BLOCK DIAGRAM .....                              | 30 |
| 5-1 PERMITTED 'MOVES' OF THE RING ENTRY MOVER .....           | 34 |
| 5-2 THE SIX RINGS AT INITIALIZATON .....                      | 36 |
| 5-3 ILLUSTRATION OF A PAGE TABLE ENTRY .....                  | 38 |
| 5-4 TOP LEVEL BLOCK DIAGRAM OF THE RING ENTRY MOVER .....     | 47 |

|      |                                                |    |
|------|------------------------------------------------|----|
| 5-5  | EXPANSION OF THE REM BLOCK .....               | 48 |
| 5-6  | MOVE_FILL SUPERVISORY STATE MACHINE .....      | 50 |
| 5-7  | RMC MOVE_FILL FLOW .....                       | 52 |
| 5-8  | HOST MOVE_FILL FLOW .....                      | 53 |
| 5-9  | AM MOVE_FILL FLOW .....                        | 54 |
| 5-10 | MOVE_FREE SUPERVISORY STATE MACHINE FLOW ..... | 56 |
| 5-11 | RMC MOVE_FREE FLOW .....                       | 58 |
| 5-12 | HOST MOVE_FREE FLOW .....                      | 59 |
| 5-13 | AM MOVE_FREE FLOW .....                        | 60 |
| 5-14 | DIAGRAM OF THE ARBITER STATE MACHINE .....     | 61 |
| 5-15 | ARBITER FLOW .....                             | 62 |
| 6-1  | RMC TIMING EXAMPLE .....                       | 69 |
| 6-2  | RMC INTERFACE BLOCK DIAGRAM .....              | 70 |
| 6-3  | REM Force Transmit Priority Algorithm .....    | 71 |
| 7-1  | ESP BLOCK CONTEXT .....                        | 73 |
| 7-2  | ESP BLOCK DIAGRAM .....                        | 74 |
| 7-3  | SYNCHRONIZER BLOCK DIAGRAM .....               | 75 |
| 7-4  | BASIC SYNCHRONIZER TIMING .....                | 76 |
| 7-5  | CSR CYCLE FORMAT .....                         | 77 |
| 7-6  | PTE CYCLE FORMAT .....                         | 77 |
| 7-7  | CSR I/F FUNCTIONAL BLOCK DIAGRAM .....         | 78 |
| 7-8  | CSR COMMAND BLOCK TIMING .....                 | 79 |
| 9-1  | PMT BLOCK DIAGRAM .....                        | 98 |

## TABLES

|     |                                                                              |    |
|-----|------------------------------------------------------------------------------|----|
| 1   | REVISION HISTORY .....                                                       | vi |
| 2-1 | PMC CONTEXT INFORMATION DICTIONARY .....                                     | 8  |
| 3-1 | RMC RECEIVE - PMC CYCLE COUNT .....                                          | 13 |
| 3-2 | RMC TRANSMIT - PMC CYCLE COUNT .....                                         | 14 |
| 3-3 | MIF CONTEXT INFORMATION DICTIONARY .....                                     | 17 |
| 5-1 | REM REGISTERS WITH RING POINTERS .....                                       | 37 |
| 5-2 | PTE ASSIGNMENTS FOR FILL AND FREE SPACE AT INITIALIZATION .....              | 39 |
| 5-3 | REM PACKET_TYPE ALLOCATION REGISTERS AND DISCARD<br>REGISTERS .....          | 43 |
| 5-4 | PTEs MAKE THE FOLLOWING MOVES BETWEEN RINGS DURING A<br>MOVE_FILL .....      | 51 |
| 5-5 | PTEs MAKE THE FOLLOWING MOVES BETWEEN RINGS DURING THE<br>FREE PROCESS ..... | 57 |
| 8-1 | PMC CSR REGISTER ALLOCATION .....                                            | 81 |

## Preface

This document defines the design of the Second Pass PMC Gate Array for the XFA Adapter. It does so in the following manner:

- \* Section 1: Executive Summary
- \* Section 2: Verbal Descriptions (Being a detailed English language description of the gate array)
  - o The MIF (Memory Interface and Packet Buffer Memory)
  - o The REM (Ring Entry Mover)
  - o The PMT (Packet Memory Test)
  - o The RMC interface
  - o The ESP interface
- \* Section 3: Hardware Descriptions (Being a functional HILO † Hardware Description Language model of the gate array)  
These are not included due to the sheer size of them. If you require to reference them, use the current or archived design base.

## ASSOCIATED DOCUMENTS

- DIGITAL: DEMNA Port Specification, Revision 4.0, Jan-1989.
- DIGITAL: XI Data Link Architecture Specification, Version X0.1.2, Apr-1987
- DIGITAL: Calypso Memory Interconnect (XMI), Revision 1.4, 28-Nov-1988.
- DIGITAL Network Architecture CSMA/CD Data Link Functional Specification, T2.0.0, 31-Mar-1989.
- DIGITAL NI Node Product Architecture Specification, T2.0.1, 11-Nov-1988.
- DIGITAL Network Architecture Maintenance Operations Functional Specification, T4.0.0, 28-Jan-1988.
- ANSI FDDI - PMD Standard, Rev 7.3, May 20, 1988.
- ANSI FDDI - PHY Standard, Rev 15, September 1, 1987.
- ANSI FDDI - MAC Standard, Rev DRAFT 7/1/87.
- ANSI FDDI - SMT Standard, Rev 5.1 September 1989.
- IEEE STD 802.2 Logical Link Control, 1985.
- DIGITAL: Clock and Data Conversion Transmitter Chip Specification, Rev 1.2, Feb 2, 1988.
- DIGITAL: Clock and Data Conversion Receiver Chip Specification, Rev 1.2, FEB 8, 1988.
- DIGITAL: Media Access Control Chip Specification, Rev 1.1, NOV. 6, 1987.
- DIGITAL: Ring Memory Controller Chip Specification, Rev 2.0, Nov 4, 1988.

---

† HILO is a simulation tool developed by GenRad Inc



- DIGITAL: Parser Sub-System Design Specification, Rev 1.0, Oct 13, 1989
- DIGITAL: XFA Module Built-In Diagnostic Functional Specification, Rev 1.0, June 30, 1989.
- DIGITAL: XFA Firmware Functional Specification, Version X3.1, Nov, 1989
- DIGITAL: XFA Hardware Functional Specification, Version 04, May, 1989

**Table 1: REVISION HISTORY**

| Date         | Revision | Summary of Changes                   |
|--------------|----------|--------------------------------------|
| 9-FEB-1989   | X??      | FIRST CUT IN DEVELOPEMENT            |
| 12-May-1989  | X01      | For DEXFA Group internal review only |
| 9-April-1990 | X01      | To update Register definitions       |
| 18-July-1990 | X20      | Revised somewhat for pass 2 design   |
| 13-Feb-1991  | X21      | Revised - Final Draft for review     |

Copyright (C) 1988, 1989, 1990,1991 by Digital Equipment Corporation

# CHAPTER 1

## EXECUTIVE SUMMARY

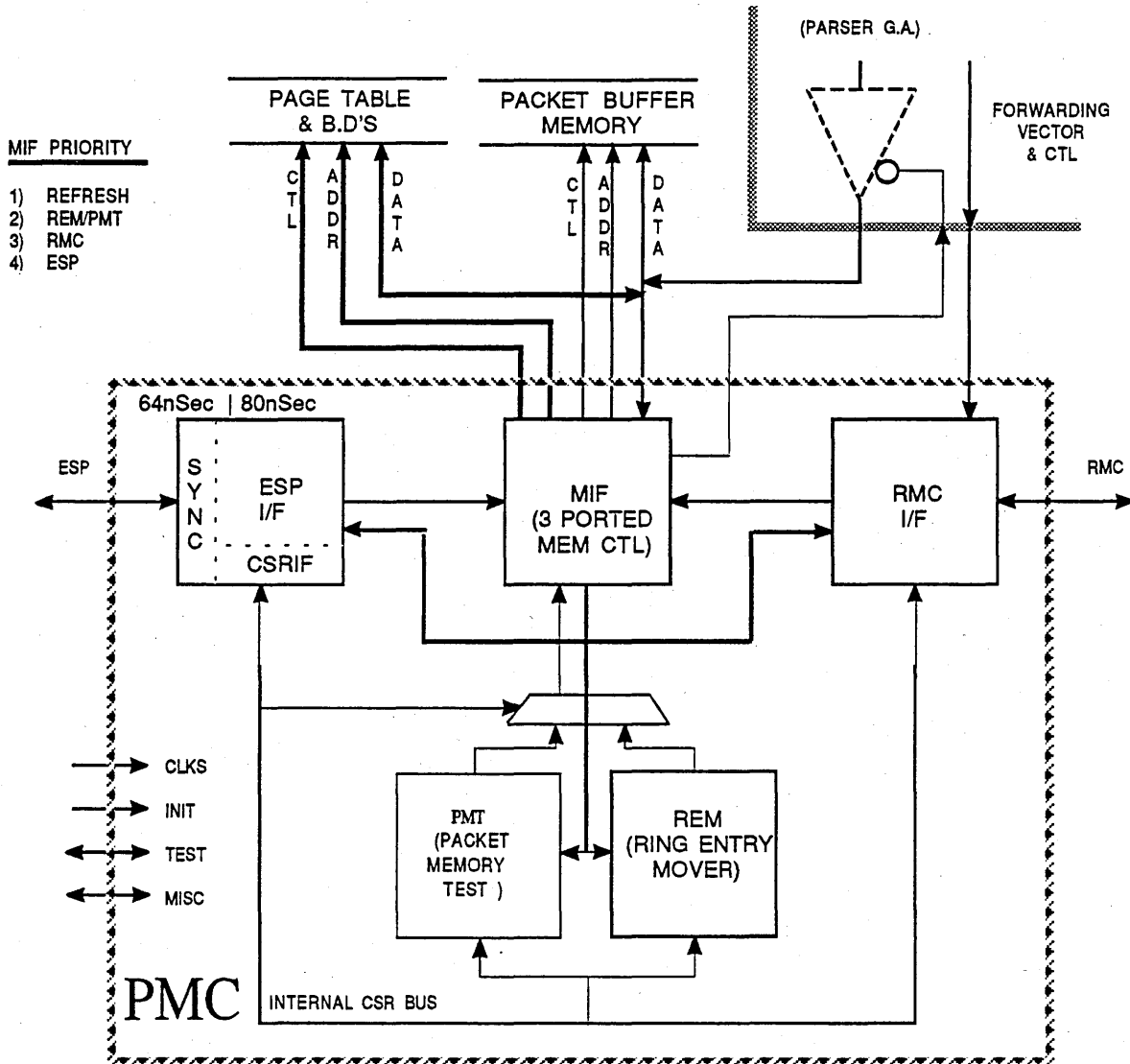
### 1.1 OVERVIEW

The Packet Memory Controller (PMC) is a logically four ported and physically three ported device with two external ports (ESP, RMC ) and one internal port shared between two functional blocks, (REM - Ring Entry Mover and PMT - Packet Memory Test). It's function is to regulate the reading and writing of the Packet Memory as required by the demands of the various interfaces. The processes of control include arbitrating access between users of the Packet Buffer and "moving" ring entries by means of manipulating the virtual address Page Table Entries.

The logical four ports of interface are the Host, Adapter Manager, RMC and Ring Entry Mover / Packet Memory Test. There is also an interface with the PARSER to enable the forwarding vector into the memory system.

The clock of the controller will be synchronized to the 80nSec FDDI clock.

Figure 1-1: PACKET MEMORY CONTROLLER - FUNCTIONAL BLOCK DIAGRAM



PMC - FUNCTIONAL BLOCK DIAGRAM

XFA\_PMC\_FBD.UIS

## 1.2 INTERFACES

### 1.2.1 RMC Interface

The RMC (Ring Memory Controller) interface is the port used by the RMC for moving FDDI packets between the packet buffer and the FDDI ring. For full details of the RMC Bus, refer to the RMC Functional Specification.

### 1.2.2 ESP Interface

The ESP interface is the port used by the ESP gate array for moving FDDI packets between host memory and the Packet Buffer. This interface is also used by the Adapter Manager; the ESP multiplexes host data and Adapter Manager data over the ESP/PMC interface bus. The interface is almost identical to the RMC interface.

## 1.3 FUNCTIONAL BLOCK DESCRIPTIONS

### 1.3.1 PACKET BUFFER MEMORY

The Packet Buffer Memory consists of 1024KBytes DRAM (expandable to 4 MB given next generation memory) implemented as a single physical bank of memory addressable by longword only. Data protection is by the use of byte parity.

### 1.3.2 PAGE TABLE AND BUFFER DESCRIPTOR MEMORY

The Page Table and B.D. Memory is 16K Longwords (expandable to 64K) of SRAM divided into specific allocations as follow: the Page Table, 14K (32K) Longwords and the Buffer Descriptors 2k (8K) Longwords. This allows the Packet Buffer to be mapped seven (four) times, ie, virtual space is seven (four) times physical space. (Numbers in parenthesis indicate that for a 4MB Packet Memory). This memory block also contains the Forwarding Vector and 'color' of each page.

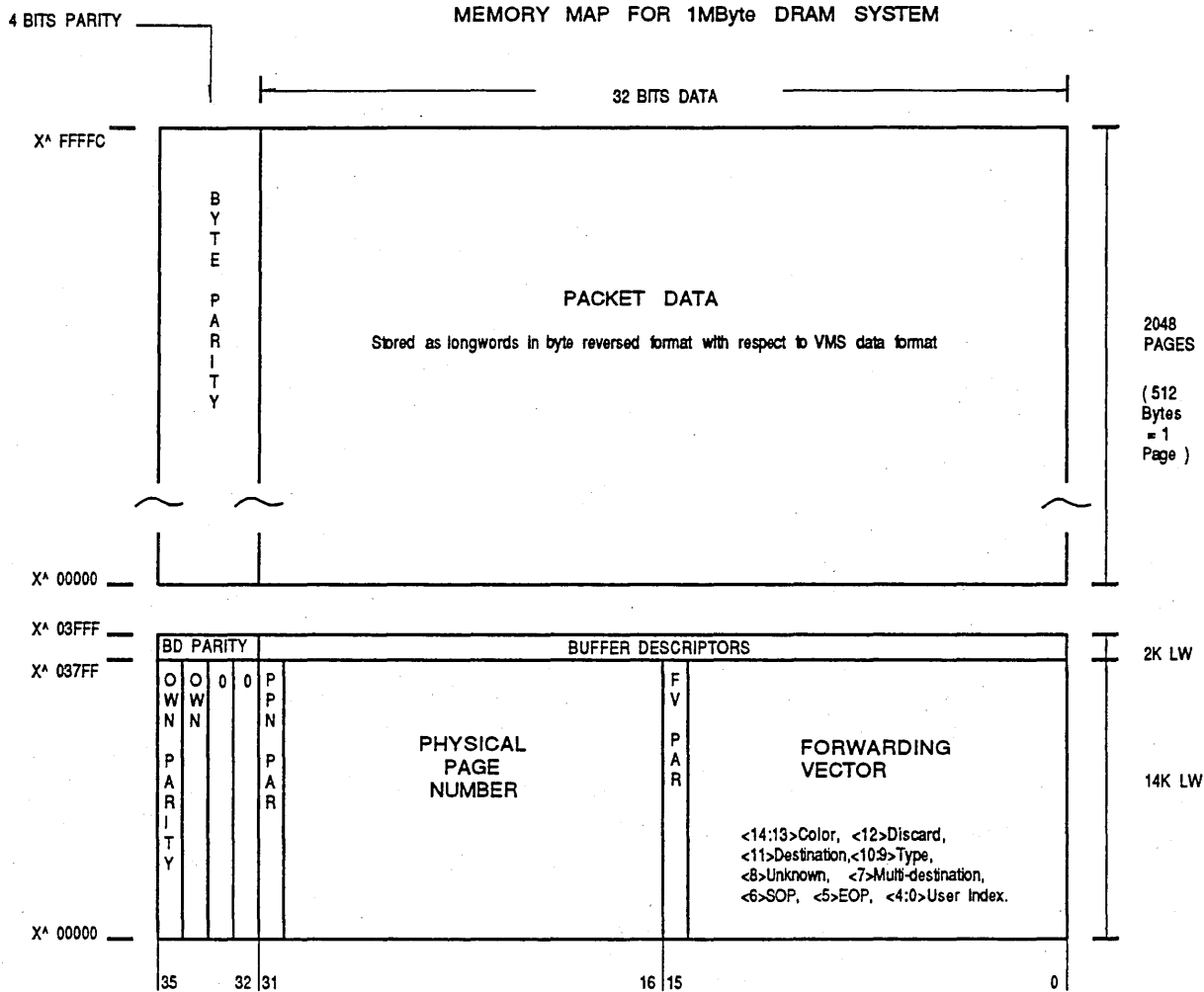
Data protection is by means of parity. The following fields are independently protected by a single bit of parity: the Physical Page Number, the Forwarding Vector and the Ownership. The Buffer Descriptors are protected by byte parity.

At initialization time, all buffers are 'colored' as belonging to one of three receive queues; the RMC\_RCV, HOST\_RCV or AM\_RCV. When buffers become free they are returned to the queue of the same color.

### 1.3.3 RING ENTRY MOVER

The Ring Entry Mover (REM) has inherent knowledge of the three external logical ports, being RMC, Adapter Manager and ESP. It interfaces with them via three pairs of rings of the form described in the RMC Functional Specification. It receives notification of all received and transmitted packets based on transmit and receive done status from the three external ports. When a packet is queued by the REM to a transmit queue, notification of the event is passed to the appropriate logical interface. The base address and length of all rings to be controlled are transferred to the REM by the Adapter Manager (via the ESP) during initialization. The REM 'moves' packets by manipulating the Page Table Entries based on information contained in a forwarding vector which is part of the Page Table Entry.

Figure 1-2: MEMORY ALLOCATION



XFA\_PMC\_PMA.UIS

### 1.3.4 PMT - Packet Memory Test

The packet memory test block is a functional block which, on power up or other initialization, will automatically test the DRAM associated with the PMC. The test can be disabled after power up such that a programmed initialization will not enable the test. The interface to the memory test is programmable in order to enable debug and manufacturing.

### 1.3.5 MIF - Memory Controller

The Memory Interface is a three port Memory Controller. There is a 'hidden' forth port, being the refresh for the DRAM. The MIF automatically takes care of the refreshing of the DRAM. Access is prioritized in the following manner:

1. Refresh
2. REM / PMT
3. RMC
4. ESP

When a memory access opportunity is present, the requester of the highest priority is granted. Logic in the RMC interface can disable ESP requests and can prioritize ESP requests over RMC requests under certain conditions. However, lockout is deliberately prevented.

### 1.3.6 RMC I/F

The RMC I/F interprets RMC Instructions and translates them into a form acceptable to the memory system. The RMC instructions are strictly as defined in the RMC Functional Specification. This interface is 80nSec synchronous.

The RMC I/F will, on an RMC Receive Clear\_Own cycle, coordinate the enabling of the forwarding vector onto the Packet Buffer Memory Data bus so that the vector is written to memory as an integral part of clearing the OWN bit.

### 1.3.7 ESP I/F

The ESP I/F is the hardware port for both the Host and the Adapter Manager. This interface synchronizes the 64nsec of the ESP to the 80nSecs of the Gate Array. It supports a superset of the RMC instructions. The differences from the RMC are as follow:

1. A read Buffer Descriptor by the Host interface for transmit (Packet Memory to Host) will result in two data cycles, not one, the first being the page table entry and the second being the Buffer Descriptor.
2. Additional commands for the Adapter Manager:
  - CSR access
  - R/W Page Table Entry

## 1.4 ADDITIONAL FEATURES

- Address translation may be enabled/disabled. This is to allow for initialization of physical page numbers, color, ownership and for debug purposes.
- The REM may be gracefully 'frozen' (after completing the current buffer or packet move).
- All HOST\_RCV packets may be forced to the Adapter Manager (by default, all packets received from the host are for the RMC).
- All RMC\_RCV packets may be forced to the Host (for debug purposes).
- All RMC\_RCV packets may be forced to the Adapter Manager (for debug purposes).

## 1.5 BYTE SWAPPING in the PMC

The RMC is always run in byte swap mode. (Byte Swap mode means that the high byte of a longword is byte 0 and the low byte is byte 3. This is to support the 68020 which understands this system.) This implies that packets are stored in packet memory in byte swap mode. In order for data to be delivered to the Host in VAX format, the PMC's ESP interface can recognize the host data stream (as opposed to the Adapter Manager's data stream) and can 'unswap' the data so it may be delivered to the host in VAX format. Note that only packet data is affected; control data structures, ie PTE's and BD's, are considered longword entities and as such maintain the same structure in both the swap and non-swap schemes.





**Table 2-1: PMC CONTEXT INFORMATION DICTIONARY**

| SIGNAL                   | DEFINITION                                                                                                                                                                                                                                  |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>RMCAD&lt;35:0&gt;</i> | RMC ADDRESS, DATA and Parity: As RMC specification                                                                                                                                                                                          |
| <i>RMCRW</i>             | RMC READ/WRITE: As RMC specification                                                                                                                                                                                                        |
| <i>RMCDTA L</i>          | RMC DATA ACKNOWLEDGE: As RMC specification                                                                                                                                                                                                  |
| <i>RMROWN</i>            | RMC OWN: As RMC specification                                                                                                                                                                                                               |
| <i>RMCRQ L</i>           | RMC REQUEST: As RMC specification                                                                                                                                                                                                           |
| <i>RMCGT L</i>           | RMC GRANT: As RMC specification                                                                                                                                                                                                             |
| <i>RMCSY L</i>           | RMC BUSY: As RMC specification                                                                                                                                                                                                              |
| <i>RMCSX</i>             | RMC START TRANSMIT: As RMC specification                                                                                                                                                                                                    |
| <i>RMCRDN</i>            | RMC RECEIVE DONE: As RMC specification                                                                                                                                                                                                      |
| <i>RMXDN</i>             | RMC TRANSMIT DONE: As RMC specification                                                                                                                                                                                                     |
| <br>                     |                                                                                                                                                                                                                                             |
| <i>PBIAD&lt;31:0&gt;</i> | ESP ADDRESS AND DATA: Equivalent to RMC specification                                                                                                                                                                                       |
| <i>PBIPAR</i>            | ESP PARITY: Longword Parity of ESPAD                                                                                                                                                                                                        |
| <i>PBIRW</i>             | ESP READ/WRITE: Equivalent to RMC specification                                                                                                                                                                                             |
| <i>PBIDTA L</i>          | ESP DATA ACKNOWLEDGE: Equivalent to RMC specification                                                                                                                                                                                       |
| <i>PBIOWN</i>            | ESP OWN: Equivalent to RMC specification                                                                                                                                                                                                    |
| <i>PBIRQ L</i>           | ESP REQUEST: Equivalent to RMC specification                                                                                                                                                                                                |
| <i>PBIGT L</i>           | ESP GRANT: Equivalent to RMC specification                                                                                                                                                                                                  |
| <i>PBISTX</i>            | ESP START TRANSMIT: Equivalent to RMC specification                                                                                                                                                                                         |
| <i>PBIRDN</i>            | ESP RECEIVE DONE: Equivalent to RMC specification                                                                                                                                                                                           |
| <i>PBIXDN</i>            | ESP TRANSMIT DONE: Equivalent to RMC specification                                                                                                                                                                                          |
| <i>PBI9AV</i>            | ESP 9 BUFFERS AVAILABLE: Status indicating that there are 9 or more free buffers in the HOST_RCV_Q                                                                                                                                          |
| <br>                     |                                                                                                                                                                                                                                             |
| <i>PMCINT</i>            | PMC INTERRUPT: A one bit status field indicating that the PMC has detected an error. The bit is passed via the ESP gate array to the 68020 subsystem resulting in a 68020 interrupt. For details of the error, read PMC CSR register 0 or 1 |
| <br>                     |                                                                                                                                                                                                                                             |
| <i>AMSTX</i>             | ADAPTER MANAGER START TRANSMIT: Equivalent to RMC specification                                                                                                                                                                             |
| <i>AMRDN</i>             | ADAPTER MANAGER RECEIVE DONE: Equivalent to RMC specification                                                                                                                                                                               |
| <i>AMXDN</i>             | ADAPTER MANAGER TRANSMIT DONE: Equivalent to RMC specification                                                                                                                                                                              |

**Table 2-1 (Cont.): PMC CONTEXT INFORMATION DICTIONARY**

| SIGNAL               | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|----------------------|-----------|----------------------|---|---|----------|-----------|---|---|-------------------|------------------|---|---|------------------------|------------------|---|---|--------------------|------------------|
| <i>XERCOD</i> <1:0>  | TRANSACTION ERROR CODE:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
|                      | <table border="1"> <thead> <tr> <th>SRAMPE</th> <th>WDGTMR</th> <th>Condition</th> <th>Action Taken by Host</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>No Error</td> <td>No Action</td> </tr> <tr> <td>1</td> <td>0</td> <td>SRAM Parity Error</td> <td>Issue Node Reset</td> </tr> <tr> <td>0</td> <td>1</td> <td>Watchdog Timer Expired</td> <td>Issue Node Reset</td> </tr> <tr> <td>1</td> <td>1</td> <td>68S/ESP Bus Broken</td> <td>Issue Node Reset</td> </tr> </tbody> </table> | SRAMPE                 | WDGTMR               | Condition | Action Taken by Host | 0 | 0 | No Error | No Action | 1 | 0 | SRAM Parity Error | Issue Node Reset | 0 | 1 | Watchdog Timer Expired | Issue Node Reset | 1 | 1 | 68S/ESP Bus Broken | Issue Node Reset |
| SRAMPE               | WDGTMR                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Condition              | Action Taken by Host |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| 0                    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | No Error               | No Action            |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| 1                    | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | SRAM Parity Error      | Issue Node Reset     |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| 0                    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Watchdog Timer Expired | Issue Node Reset     |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| 1                    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 68S/ESP Bus Broken     | Issue Node Reset     |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>MEMSIZ</i>        | MEMORY SIZE: A hardware jumper input to the gate array. When deasserted it indicates that the module has been manufactured with 256K x 4 DRAMS and 16K x 4 SRAMS. When asserted it indicates that the module has been manufactured with 1M x 4 DRAMS and 64K x 4 SRAMS.                                                                                                                                                                                                                                 |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>PKTDIS</i>        | PACKET DISCARD: a signal from the PARSER gate array indicating that the current receive packet will be discarded. The PMC will respond to all RMC receive data transactions in the normal way but will not deliver the data to packet memory. This will allow the packet memory be used for more useful purposes. Buffer descriptor and clear own cycles will be dealt with in the regular manner                                                                                                       |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>FVRDY</i>         | FORWARDING VECTOR READY: Status from the PARSER G.A. indicating that the forwarding vector is valid                                                                                                                                                                                                                                                                                                                                                                                                     |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>FVOE L</i>        | OUTPUT ENABLE FORWARDING VECTOR: A command from the PMC to the PARSER G.A. to output the forwarding vector onto the PMC data lines. Deassertion of this signal indicates 'done-ness' to the PARSER.                                                                                                                                                                                                                                                                                                     |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>OES L</i>         | Output Enable all 36 bits of SRAM data                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>WELWRD L</i>      | Write Enable low word of the SRAM                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>WEHWRD L</i>      | Write Enable high word of the SRAM                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>WEHNIB L</i>      | Write Enable high nibble of the SRAM                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>SRAMA</i> <15:0>  | 16 bits of SRAM address                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>OEDRAM L</i>      | Output Enable the DRAM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>WEDRAM L</i>      | Write Enable the DRAM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>RAMRAS L</i>      | Select the row addressed by DRAMAD                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>CAS L</i>         | Select the column addressed by DRAMAD                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>DRAMAD</i> <9:0>  | 10 bits of dram address                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |
| <i>RAMDAT</i> <35:0> | 36 bits of data shared between the PMC, SRAM, DRAM and Forwarding Vector                                                                                                                                                                                                                                                                                                                                                                                                                                |                        |                      |           |                      |   |   |          |           |   |   |                   |                  |   |   |                        |                  |   |   |                    |                  |

**Table 2-1 (Cont.): PMC CONTEXT INFORMATION DICTIONARY**

| SIGNAL                  | DEFINITION                                                                                                                                                                                                                                           |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>IVIN</i>             | Input for the IV100 cell, used in silicon characterization                                                                                                                                                                                           |
| <i>IVOUT</i>            | Output of the IV100 cell, used in silicon characterization                                                                                                                                                                                           |
| <i>PAROUT</i>           | Output of the Input Cells NAND tree used for testing all input cells                                                                                                                                                                                 |
| <i>PMCTRI</i>           | When asserted will tristate all tristateable output cells. Used to isolate the gate array from other circuitry for module level testing                                                                                                              |
| <i>SMSEL&lt;4:0&gt;</i> | A test input controlling a mux selecting any of the state machine states (and other selected signals) for test visibility and debug purposes                                                                                                         |
| <i>SM&lt;14:0&gt;</i>   | The output of a mux selecting any of the state machine states (and other selected signals) for test visibility and debug purposes                                                                                                                    |
| <i>BYTCLK</i>           | Input for a 80nSec TTL clock having a 1:1 mark/space ratio                                                                                                                                                                                           |
| <i>SYMCLK</i>           | Input for a 40nSec TTL clock having a 1:1 mark/space ratio and having every other rising edge common with the rising edge of BYTCLK                                                                                                                  |
| <i>BCLK34</i>           | Input for a 64nSec CMOS clock having a 2:3 mark/space ratio. This clock is true for phases 3 and 4 of a source (XMI) clock having 6 phases                                                                                                           |
| <i>BCLK61</i>           | Input for a 64nSec CMOS clock having a 2:3 mark/space ratio. This clock is true for phases 6 and 1 of a source (XMI) clock having 6 phases                                                                                                           |
| <i>INIT L</i>           | Input for a global initialization signal. Assertion of this signal will cause all register elements within the PMC to reset to a predictable value                                                                                                   |
| <i>FINITL</i>           | An output which will assert when any of: INIT L, XERCOD<1:0> asserts and stays true for at least 8 cycles regardless of the time of assertion of the activating signals. This signal is used to initialize the FDDI chip set and PARSER gate arrays. |

## CHAPTER 3

### MEMORY INTERFACE AND PACKET BUFFER MEMORY

#### 3.1 BANDWIDTH

The design goal of the PMC is to satisfy the bandwidth requirements of minimum sized FDDI LLC packets. Please refer to the performance analysis work of REGE / KALKUNTE for more details.

#### 3.2 DESIGN ATTRIBUTES

- Every access to DRAM is allowed one idle cycle at the beginning of each access for precharge. Where DRAM requests require address translation (SRAM access), the precharge cycle is used for that purpose.
- SRAM accesses for writing Forwarding Vectors are allowed one idle cycle before and one idle cycle after the actual write to allow for bus turn-around
- Refresh consumes consumes 3 cycles every 10.24uSecs or less than 2%

#### 3.3 Minimum Sized Forwarded Packets

There are two critical bandwidth requirements: to sustain minimum packets and 33 byte (in PMC) packets (ie smallest packet with two RMC data bursts).

Following is a breakdown of a minimum sized LLC packet. Also shown is the relationship between a 33 byte 'in PMC' packet and the 'on the wire' time.

Figure 3-1: MINIMUM LLC PACKET CYCLE COUNT

MINIMUM 'ON THE WIRE' LLC PACKET

|                      |    |                            |
|----------------------|----|----------------------------|
| Interpacket Gap      | 8  | Byte times                 |
| Starting delimiter   | 1  |                            |
| Frame Control        | 1  |                            |
| Destination Address  | 6  |                            |
| Source Address       | 6  |                            |
| DSAP                 | 1  |                            |
| SSAP                 | 1  |                            |
| Control              | 1  |                            |
| Data                 | 0  |                            |
| Frame Check Sequence | 4  |                            |
| Terminator           | ½  |                            |
| Status               | 1½ |                            |
| -----                |    |                            |
|                      |    | 31 Byte times (2.48 uSecs) |
| =====                |    |                            |

MINIMUM 'IN PMC' LLC PACKET

|                      |   |                        |
|----------------------|---|------------------------|
| Packet Header        | 3 |                        |
| Frame Control        | 1 |                        |
| Destination Address  | 6 |                        |
| Source Address       | 6 |                        |
| DSAP                 | 1 |                        |
| SSAP                 | 1 |                        |
| Control              | 1 |                        |
| Data                 | 0 |                        |
| Frame Check Sequence | 4 |                        |
| -----                |   |                        |
|                      |   | 23 Bytes (6 Longwords) |
| =====                |   |                        |

THE '33 BYTE DATA CASE'

'On the wire' size = 41 byte times = 3.28 uSecs

'In PMC' size = 9 longwords

### 3.4 CYCLE COUNTS

Following are memory cycle counts for a minimum sized LLC packet being received / transmitted from / to the RMC.

**Table 3-1: RMC RECEIVE - PMC CYCLE COUNT**

| INTERFACE | CYCLE TYPE | CYCLE COUNT                                     |
|-----------|------------|-------------------------------------------------|
| RMC       | XLATE      | 1                                               |
|           | RAS        | 1                                               |
|           | CAS / W    | 1                                               |
|           | 5 W        | 5 (8 cycle access)                              |
|           | XLATE      | 1                                               |
|           | W BD       | 1 (2 cycle access)                              |
|           | IDLE       | 1 (Bus turn around for FV inst)                 |
|           | C O        | 1                                               |
|           | IDLE       | 1 (Bus turn around for FV inst. 3 cycle access) |
|           | REM        | R PTE                                           |
| W PTE     |            | 1 (1 cycle access)                              |
| ESP       | XLATE      | 1 (Address indirection cached)                  |
|           | R BD       | 1 (2 cycle access)                              |
|           | XLATE      | 1                                               |
|           | RAS        | 1                                               |
|           | CAS / R    | 1                                               |
|           | 5 R        | 5 (8 cycle access)                              |
| REM       | C O        | 1 (1 cycle access)                              |
|           | R PTE      | 1 (1 cycle access)                              |
| REM       | W PTE      | 1 (1 cycle access)                              |
|           |            | —                                               |
|           |            | 28 CYCLES = 2.24 uSecs                          |
|           |            | ==                                              |

**Table 3-2: RMC TRANSMIT - PMC CYCLE COUNT**

| INTERFACE | CYCLE TYPE | CYCLE COUNT                                      |                    |
|-----------|------------|--------------------------------------------------|--------------------|
| ESP       | XLATE      | 1 (Address indirection cached)                   |                    |
|           | RAS        | 1                                                |                    |
|           | CAS / W    | 1                                                |                    |
|           | 5 W        | 5 (8 cycle access)                               |                    |
|           | XLATE      | 1                                                |                    |
|           | W BD       | 1 (2 cycle access)                               |                    |
|           | C O        | 1 (1 cycle access)                               |                    |
|           | REM        | R PTE                                            | 1 (1 cycle access) |
|           |            | W PTE                                            | 1 (1 cycle access) |
| RMC       | XLATE      | 1 (Address indirection cached)                   |                    |
|           | R BD       | 1 (2 cycle access)                               |                    |
|           | XLATE      | 1                                                |                    |
|           | RAS        | 1                                                |                    |
|           | CAS / R    | 1                                                |                    |
|           | 5 R        | 5 (8 cycle access)                               |                    |
|           | XLATE      | 0                                                |                    |
|           | W BD       | 0 (PMC can be programmed to ignore RMC_XMT_W_BD) |                    |
|           | C O        | 1 (1 cycle access)                               |                    |
| REM       | R PTE      | 1 (1 cycle access)                               |                    |
|           | W PTE      | 1 (1 cycle access)                               |                    |
|           | -          |                                                  |                    |
|           |            | 26 CYCLES = 2.08 uSecs                           |                    |
|           |            | ==                                               |                    |

Based on the same instructions as above, the 33 byte PMC data case requires 38 and 36 cycles for RMC receive and transmit respectively. Based solely on available cycles, it is clear that the memory system can move packets faster than FDDI can transmit / receive them. However, this does not take into account other factors such as burstiness and queuing effects. Refer to the performance analysis work of REGE / KALKUNTE for more information in this matter. The conclusion of that analysis is that infinite streams of minimum sized LLC packets can be transmitted / received by the PMC but due to Host Data path limitations, the DEMFA / Host as a system cannot.

### 3.5 MIF REQUIRED INSTRUCTIONS

The following instructions are internally generated. It should also be noted that the MIF will implement the Refresh logic.

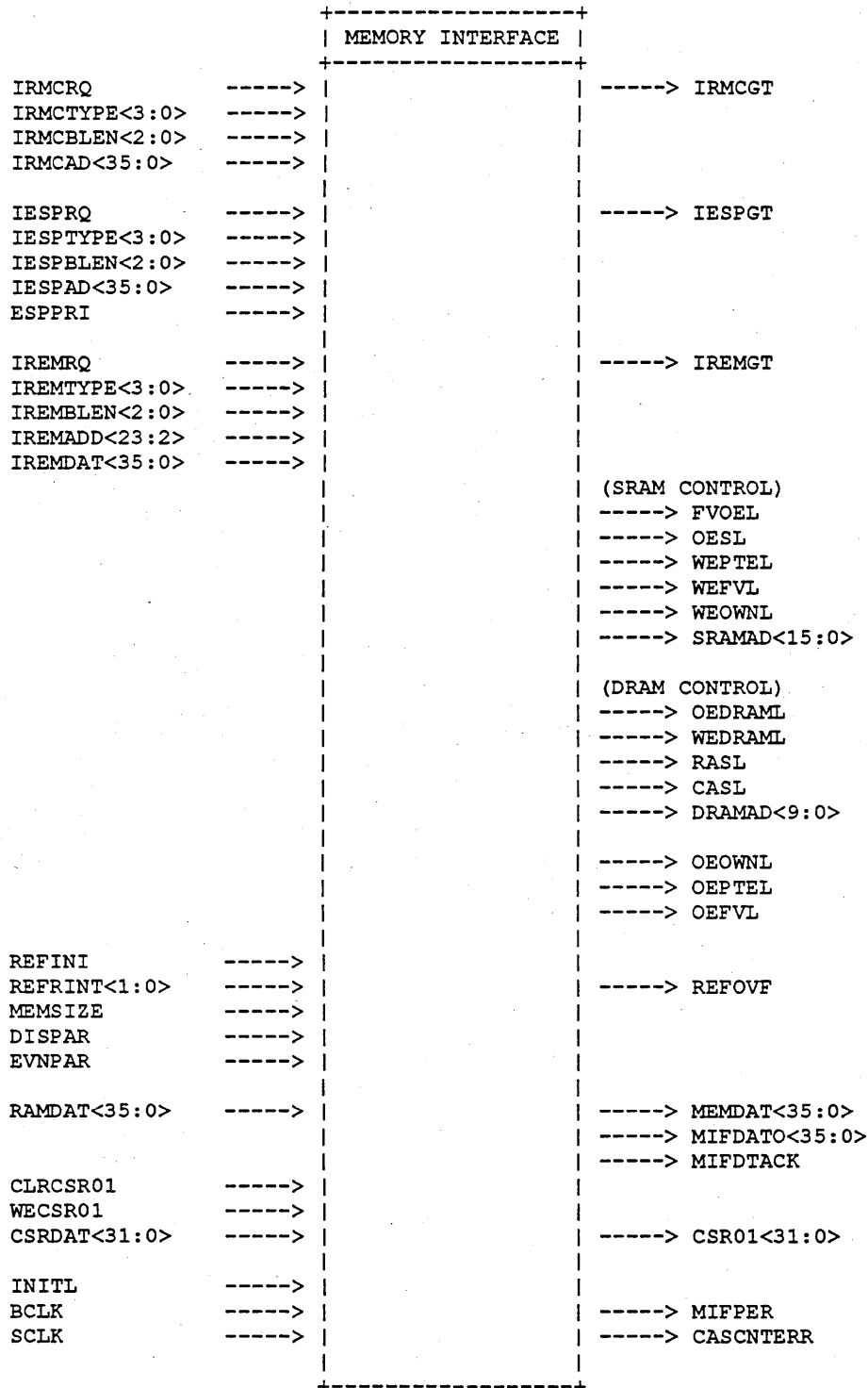
**Figure 3-2: MIF SUPPORTED CYCLE TYPES**

| INSTRUCTION # (^B) | INSTRUCTION                                          |
|--------------------|------------------------------------------------------|
| 0000               | DRAM READ BURST WITH XLATE                           |
| 0001               | DRAM READ BURST, NO XLATE                            |
| 0010               | DRAM WRITE BURST WITH XLATE                          |
| 0011               | DRAM WRITE BURST, NO XLATE                           |
| 0100               | READ BUFFER DESCRIPTOR WITH XLATE (implied PTE read) |
| 0101               | READ BUFFER DESCRIPTOR, NO XLATE                     |
| 0110               | WRITE BUFFER DESCRIPTOR WITH XLATE                   |
| 0111               | WRITE BUFFER DESCRIPTOR, NO XLATE                    |
| 1000               | Not used                                             |
| 1001               | Not used                                             |
| 1010               | Not used                                             |
| 1011               | Not used                                             |
| 1100               | READ PTE                                             |
| 1101               | WRITE PTE (WITH IMPLIED SET OWN)                     |
| 1110               | CLEAR OWN                                            |
| 1111               | CLEAR OWN and WRITE FORWARDING VECTOR                |
| XXXX               | REFRESH                                              |



### 3.6 MIF CONTEXT

Figure 3-3: MIF CONTEXT



**Table 3-3: MIF CONTEXT INFORMATION DICTIONARY**

| SIGNAL                     | DEFINITION                                                                                                                                                                                                                        |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>IRMCRQ</i>              | An internal version of the RMC Request line                                                                                                                                                                                       |
| <i>IRMCTYPE&lt;3:0&gt;</i> | A four bit field defining the cycle requested. See Figure 3-2.                                                                                                                                                                    |
| <i>IRMCBLEN&lt;2:0&gt;</i> | A three bit field defining the length of a DRAM burst access. Length of ^B000 = 1 longword, ^B111=8 longwords. Length field is ignored for SRAM accesses. This is identical to bits <31:29> of the RMC/ESP Command/Address field. |
| <i>IRMCAD&lt;35:0&gt;</i>  | A 36 bit (32 data + byte parity) bus inputting data or address information from the RMC interface. Note that for an address, parity is not necessarily valid                                                                      |
| <i>IRMCGT</i>              | The MIF's response to an RMC request                                                                                                                                                                                              |
| <i>IESPRQ</i>              | An internal version of the ESP Request line                                                                                                                                                                                       |
| <i>IESPTYPE&lt;3:0&gt;</i> | A four bit field defining the cycle requested. See Figure 3-2.                                                                                                                                                                    |
| <i>IESPBLN&lt;2:0&gt;</i>  | A three bit field defining the length of a DRAM burst access. Length of ^B000 = 1 longword, ^B111=8 longwords. Length field is ignored for SRAM accesses.                                                                         |
| <i>IESPAD&lt;35:0&gt;</i>  | A 36 bit (32 data + byte parity) bus inputting data or address information form the ESP interface. Note that for an address, parity is not valid                                                                                  |
| <i>ESPPRI</i>              | An input requiring that the ESP be prioritized over the RMC, the opposite of the normal priority for ESP / RMC                                                                                                                    |
| <i>IESPGT</i>              | The MIF's response to an ESP request                                                                                                                                                                                              |
| <i>IREMRQ</i>              | Request from the REM for service                                                                                                                                                                                                  |
| <i>IREMTYPE&lt;3:0&gt;</i> | A four bit field defining the cycle requested. See Figure 3-2.                                                                                                                                                                    |
| <i>IREMBLEN&lt;2:0&gt;</i> | A three bit field defining the length of a DRAM burst access. Length of ^B000 = 1 longword, ^B111=8 longwords. Length field is ignored for SRAM accesses.                                                                         |
| <i>IREMADD&lt;23:2&gt;</i> | A 22 bit bus inputting address information from the Ring Entry Mover.                                                                                                                                                             |
| <i>IREMDAT&lt;35:0&gt;</i> | A 36 bit (32 data + byte parity) bus inputting data from the Ring Entry Mover.                                                                                                                                                    |
| <i>IREMGT</i>              | The MIF's response to an REM request                                                                                                                                                                                              |
| <i>FVOEL</i>               | A signal which ultimately goes to the Parser Gate Array to output the Forwarding Vector onto the PMC's RAMDAT bus                                                                                                                 |
| <i>OESL</i>                | A signal which controls the outputting of data from all the external SRAMs onto the PMC's RAMDAT bus                                                                                                                              |
| <i>WEOWNL</i>              | A signal which controls the writing of data into the external SRAMs connected to bits <35:32> of the PMC's RAMDAT bus                                                                                                             |
| <i>WEPTL</i>               | A signal which controls the writing of data into the external SRAMs connected to bits <31:16> of the PMC's RAMDAT bus                                                                                                             |
| <i>WEFVL</i>               | A signal which controls the writing of data into the external SRAMs connected to bits <15:0> of the PMC's RAMDAT bus                                                                                                              |
| <i>SRAMAD&lt;15:0&gt;</i>  | A 16-bit bus providing all the external SRAMs with address information                                                                                                                                                            |
| <i>OEDRAML</i>             | A signal which controls the outputting of data from all the external DRAMs onto the PMC's RAMDAT bus                                                                                                                              |
| <i>WEDRAML</i>             | A signal which controls the writing of data into all the external DRAMs, ie, bits <35:0> of the PMC's RAMDAT bus                                                                                                                  |
| <i>RASL</i>                | A signal which controls the row address select of all the external DRAMs                                                                                                                                                          |
| <i>CASL</i>                | A signal which controls the column address select of all the external DRAMs                                                                                                                                                       |
| <i>DRAMAD&lt;9:0&gt;</i>   | A 10-bit bus providing all the external DRAMs with row and column address information                                                                                                                                             |

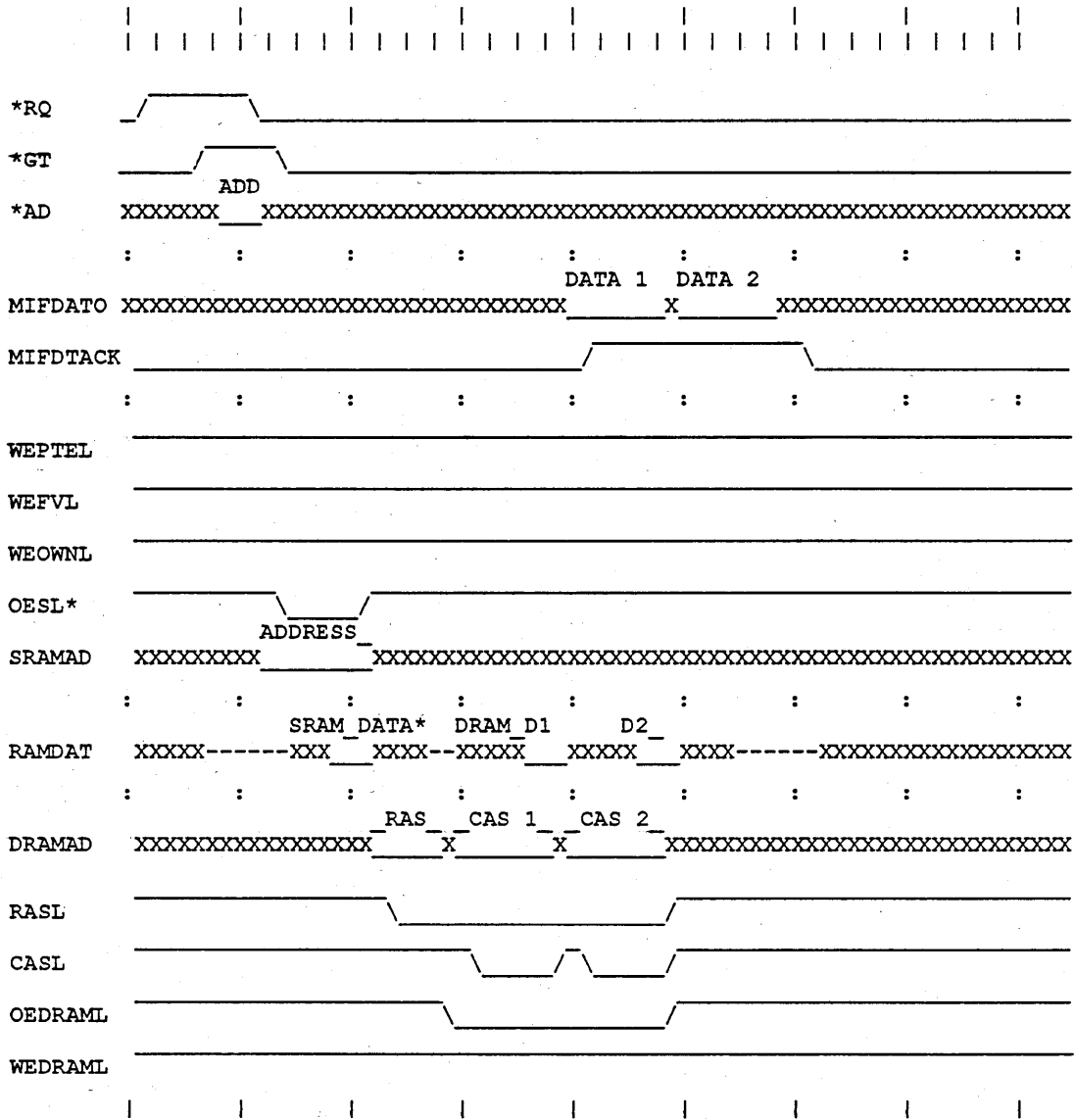
**Table 3-3 (Cont.): MIF CONTEXT INFORMATION DICTIONARY**

| SIGNAL                     | DEFINITION                                                                                                                                                                                                                                                                                                                                        |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>OEOWNL</i>              | A signal which controls the outputting of data bits <35:32> from the PMC gate array to the the PMC's RAMDAT bus                                                                                                                                                                                                                                   |
| <i>OEPTL</i>               | A signal which controls the outputting of data bits <31:16> from the PMC gate array to the the PMC's RAMDAT bus                                                                                                                                                                                                                                   |
| <i>OEVL</i>                | A signal which controls the outputting of data bits <15:0> from the PMC gate array to the the PMC's RAMDAT bus                                                                                                                                                                                                                                    |
| <i>REFINI</i>              | A signal, for testability purposes, which resets the refresh counters to their initialized value                                                                                                                                                                                                                                                  |
| <i>REFOVF</i>              | A signal, for testability purposes, indicates that the refresh counters have overflowed. The time between the deassertion of REFINI and the assertion of REFOVF indicates the time it takes to completely refresh the DRAMs                                                                                                                       |
| <i>REFRINT&lt;1:0&gt;</i>  | Two bits of information provided through a CSR indicating the refresh interval for the DRAM. ^B11 = 5.12uSecs, ^B10 = 10.24uSecs, ^B01 = 20.48uSecs, ^B00 = 40.96uSecs.                                                                                                                                                                           |
| <i>MEMSIZE</i>             | Information provided through a CSR indicating the physical size of the DRAM. ^B0 = 1MB, ^B1 = 4MB                                                                                                                                                                                                                                                 |
| <i>DISPAR</i>              | A signal which, when asserted, disables parity checking                                                                                                                                                                                                                                                                                           |
| <i>EVNPAR</i>              | A signal which, when asserted, forces parity generators and checkers to generate / check for even parity instead of the default odd parity                                                                                                                                                                                                        |
| <i>RAMDAT&lt;35:0&gt;</i>  | A 36 bit (32 bit data and byte parity or 33 bit data and 3 bits of parity in the case of a PTE access) bus carrying information from the external DRAMs or SRAMs                                                                                                                                                                                  |
| <i>MEMDAT&lt;35:0&gt;</i>  | A 36 bit (32 bit data and byte parity or 33 bit data and 3 bits of parity in the case of a PTE access) bus carrying information from the MIF destined for the external DRAMs or SRAMs                                                                                                                                                             |
| <i>MIFDATO&lt;35:0&gt;</i> | A 36 bit (32 bit data and byte parity or 33 bit data and 3 bits of parity in the case of a PTE access) bus carrying information requested from the MIF to all of the possible requestors being the RMCIF, ESPIF, REM or PMT                                                                                                                       |
| <i>MIFDTACK</i>            | A one bit status from MIF indicating either "I have accepted data" (for a write operation) or "The data I am giving you is valid" for a read operation. This signal, like MIFDATO, is provided to all the possible requestors. It is up to the external requestor to determine if the MIFDTACK is associated with the external requestors request |
| <i>CLRCSR01</i>            | A signal which requires CSR01 to be reset to its initialization value                                                                                                                                                                                                                                                                             |
| <i>WECSR01</i>             | A signal which enables the writing of CSR01 with CSRDAT information, for testability purposes                                                                                                                                                                                                                                                     |
| <i>CSRDAT&lt;31:0&gt;</i>  | 32 bits of data which, when WECSR01 is true, will be written into CSR01                                                                                                                                                                                                                                                                           |
| <i>CSR01&lt;31:0&gt;</i>   | 32 bits of data which comprise CSR01                                                                                                                                                                                                                                                                                                              |
| <i>INITL</i>               | A global initialization signal of true low assertion which, when asserted, resets all data storage elements to their predetermined initialization values                                                                                                                                                                                          |
| <i>BCLK</i>                | A clock of 80nSec nominal period having 1:1 mark / space ratio                                                                                                                                                                                                                                                                                    |
| <i>SCLK</i>                | A clock of 40nSec nominal period having 1:1 mark / space ratio and synchronized with BCLK. The phase relationship to BCLK is that each rising edge of SCLK is nominally concurrent with the rising or falling edge of BCLK                                                                                                                        |

### 3.7 MIF TIMING

Figure 3-4: DRAM READ BURST

INSTRUCTIONS ^B0000 AND ^B0001 (2 cycle access shown)



^^^^^^  
 \* Only ^B0000 accesses SRAM in this cycle

Figure 3-5: DRAM WRITE BURST

INSTRUCTIONS ^B0010 AND ^B0011 (2 cycle access shown)

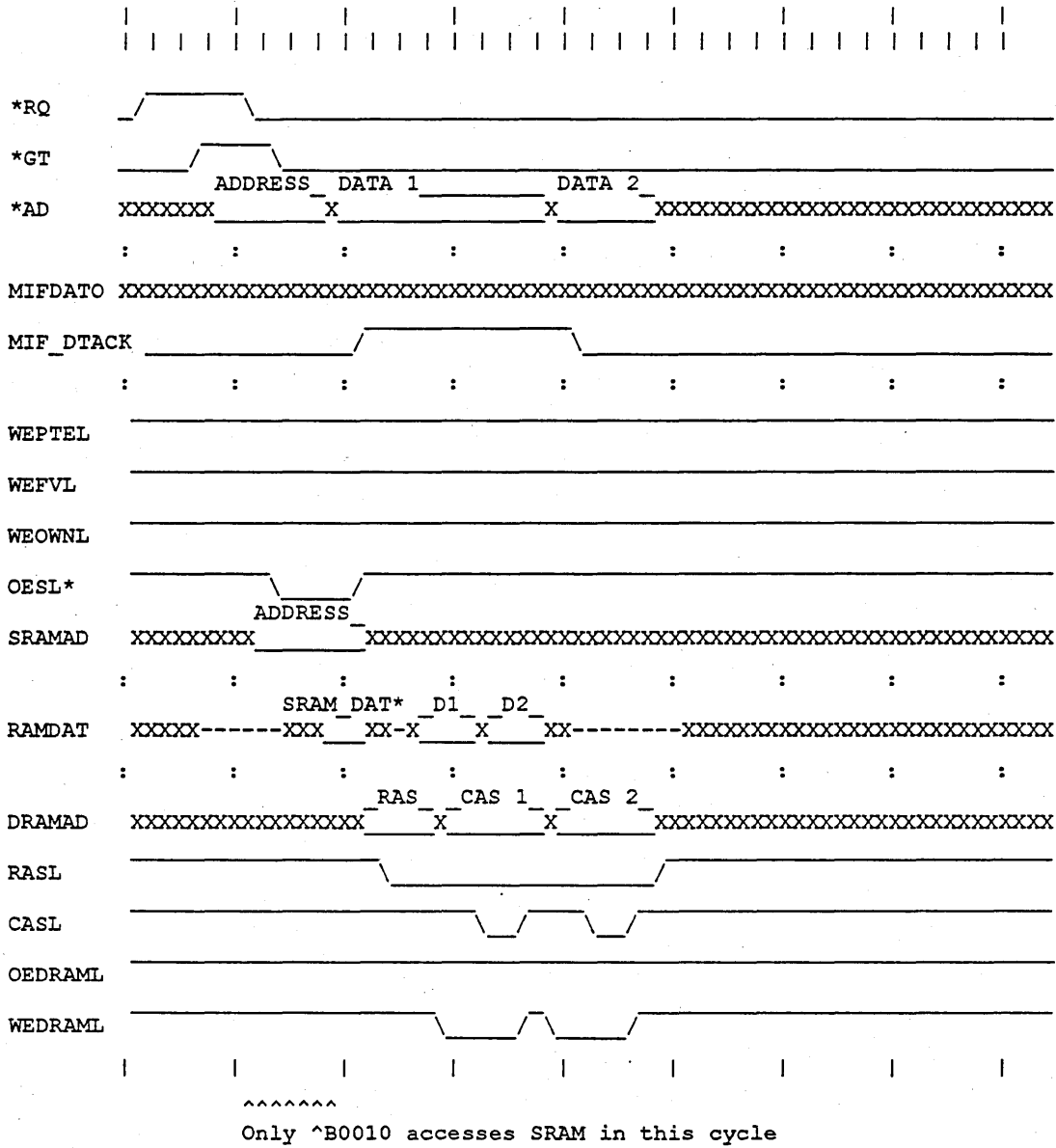


Figure 3-6: READ BUFFER DESCRIPTOR

INSTRUCTIONS ^B0100 AND ^B0101

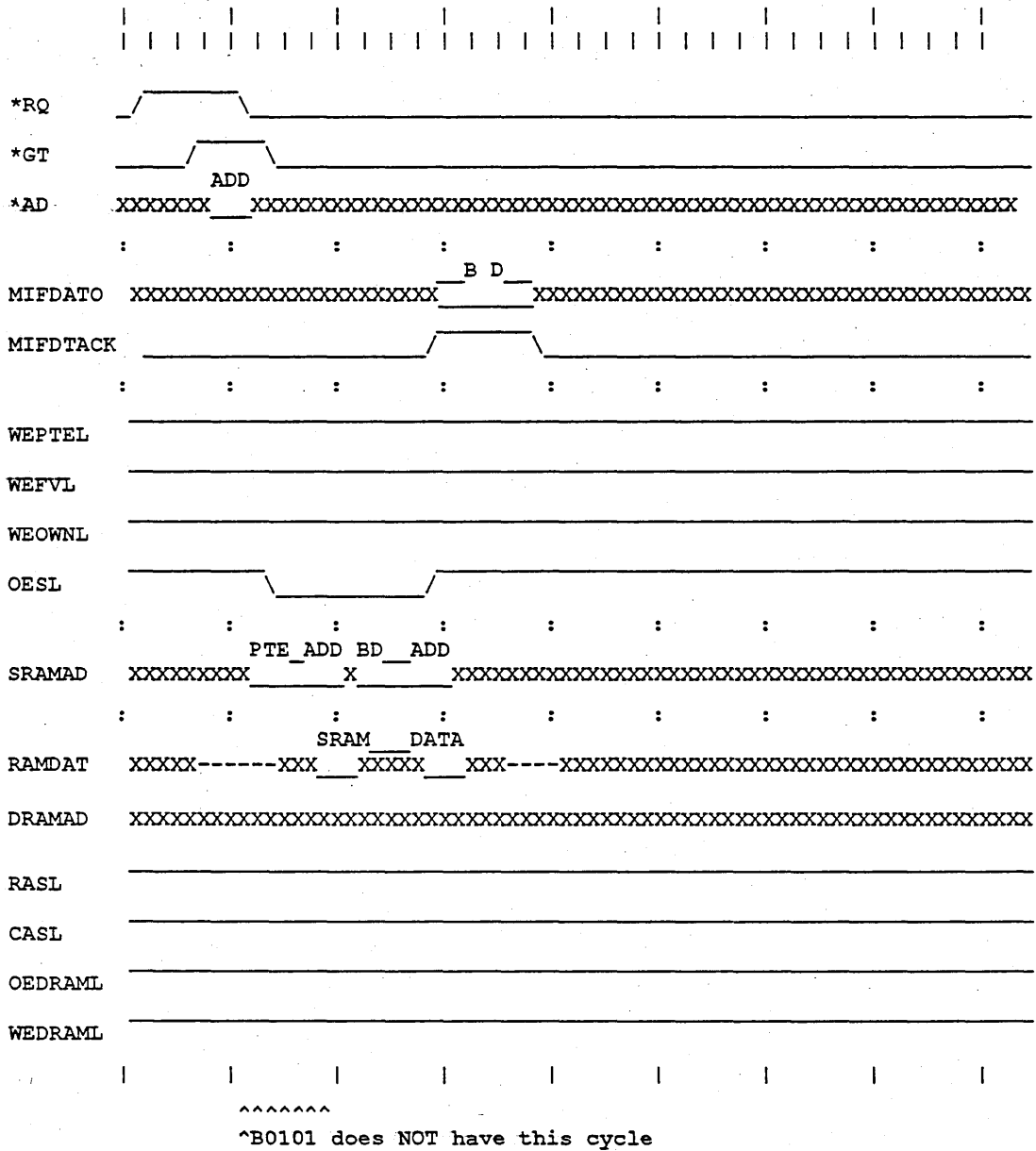


Figure 3-7: WRITE BUFFER DESCRIPTOR

INSTRUCTIONS ^B0110 AND ^B0111

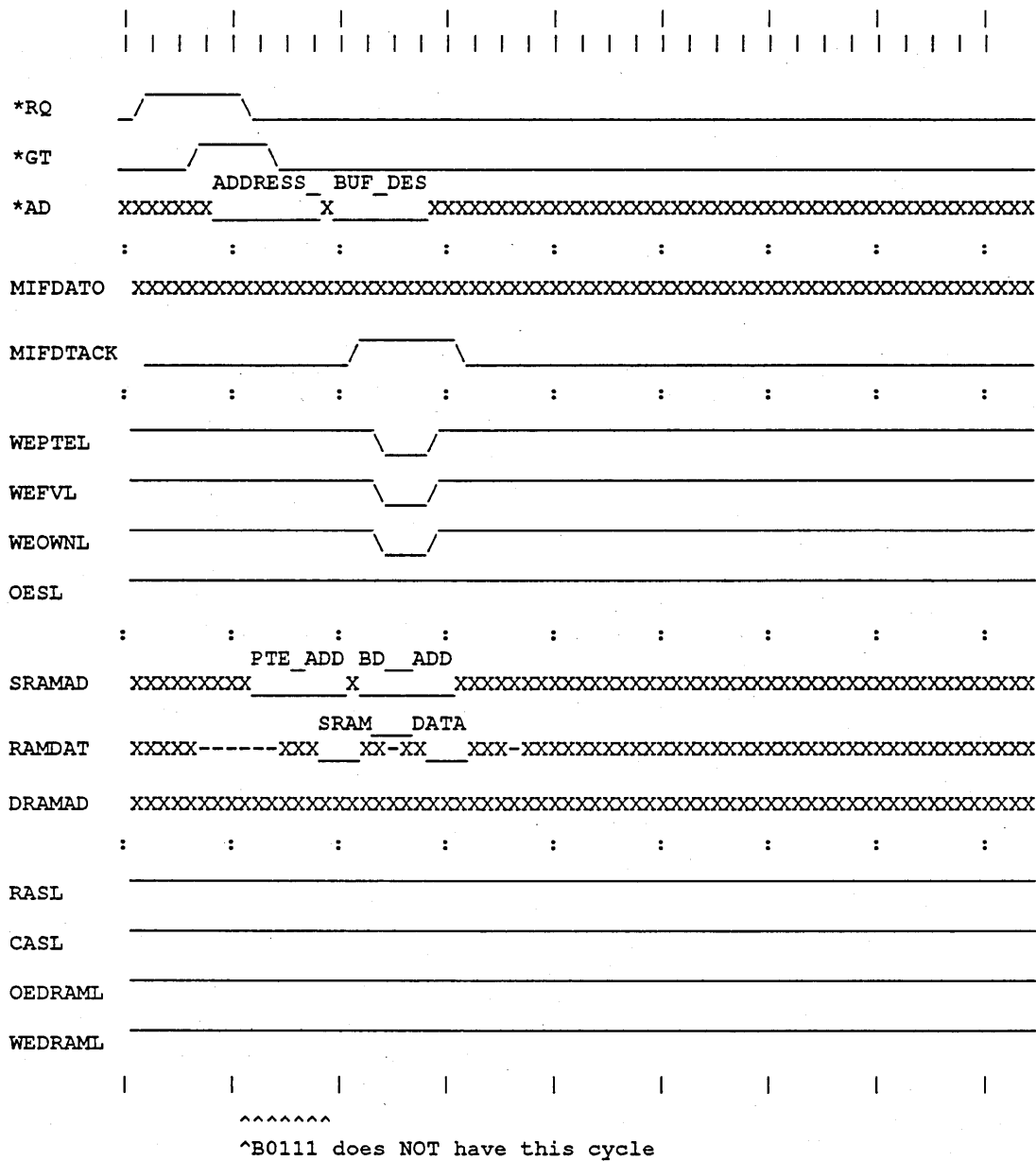








Figure 3-10: WRITE FORWARDING VECTOR, CLEAR OWN

INSTRUCTIONS ^B1111

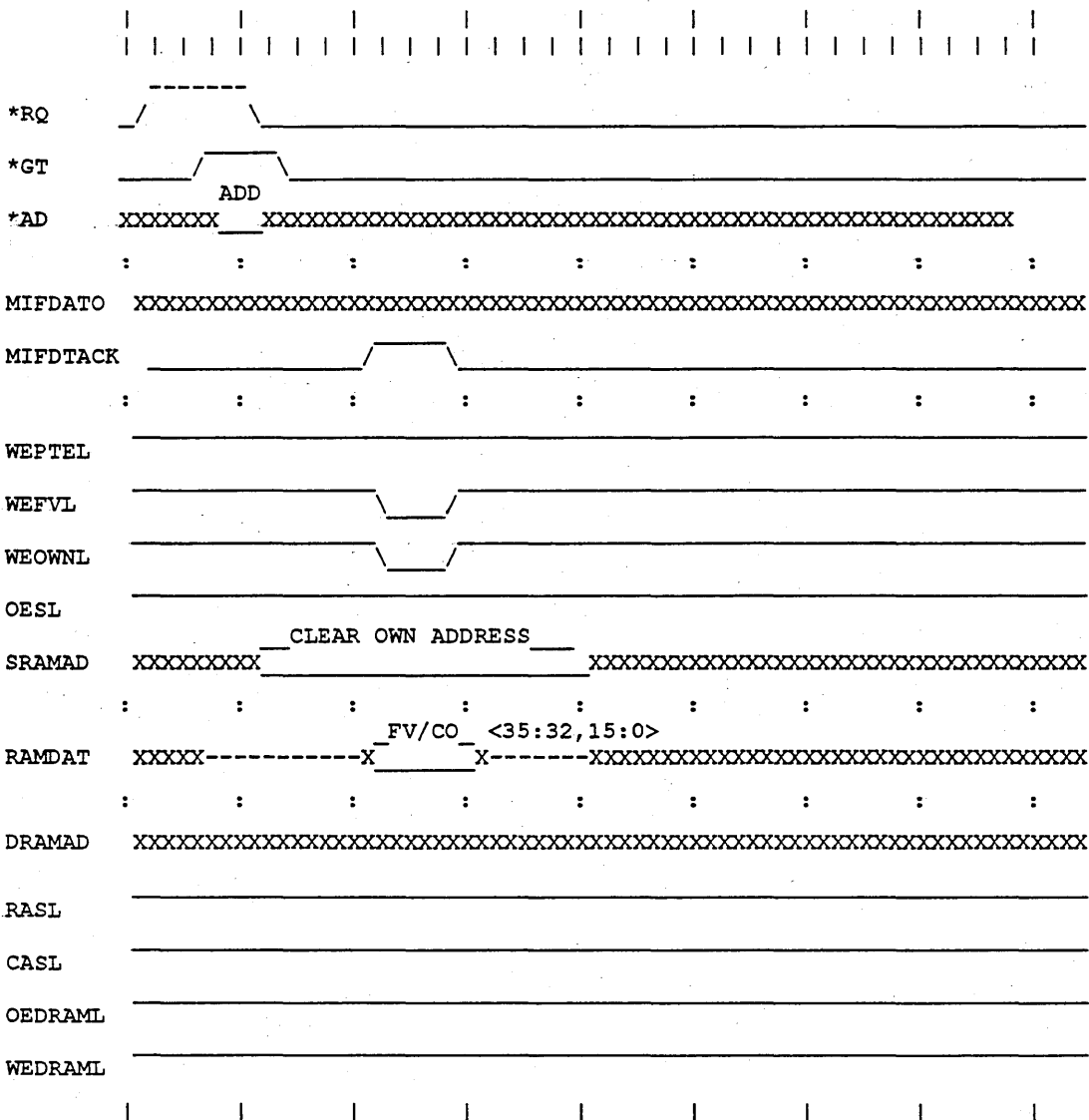
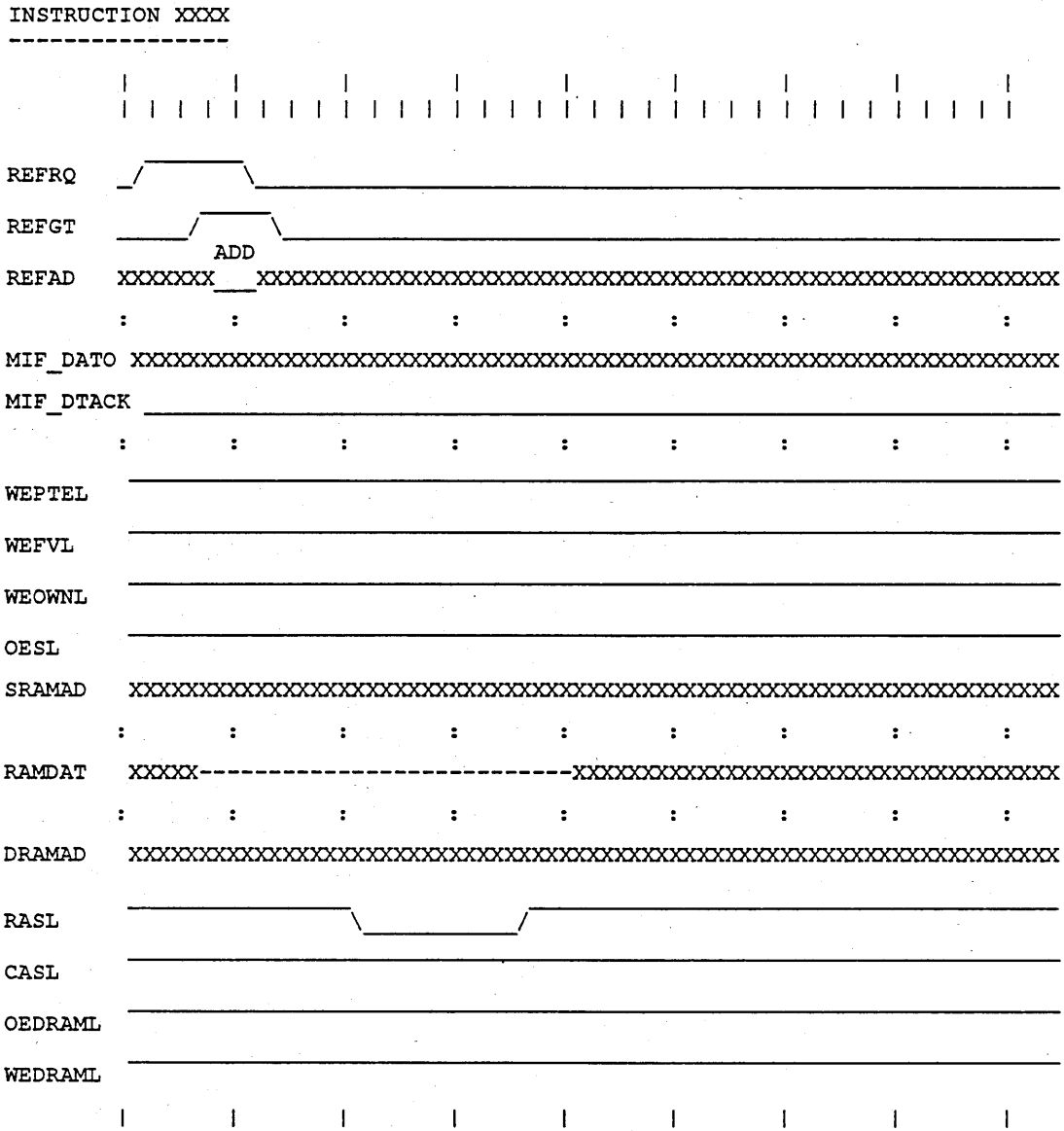
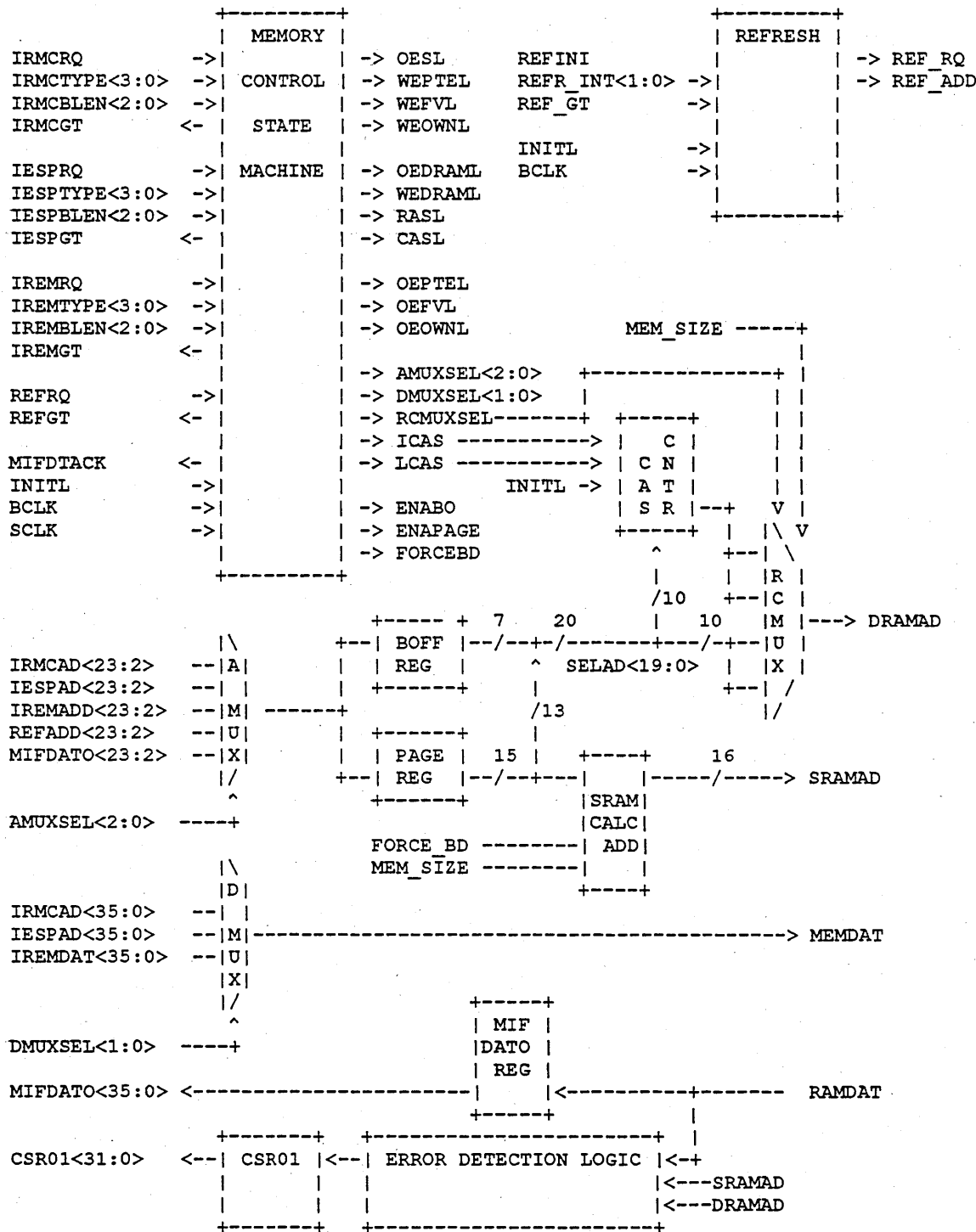


Figure 3-11: REFRESH



### 3.8 EXPANSION OF MIF FUNCTIONALITY

Figure 3-12: MIF FUNCTIONAL BLOCK DIAGRAM



### 3.8.1 MEMORY CONTROL STATE MACHINE

This state machine controls the timing of all SRAM and DRAM transactions. The timing is according to the timing diagrams shown in this chapter.

### 3.8.2 MIF ADDRESS MUX

This mux controls the source address for the memories. They are encoded as follow:

- ^B111 - Illegal
- ^B110 - Illegal
- ^B101 - Illegal
- ^B100 - MIF\_DATO
- ^B011 - REF\_ADD
- ^B010 - IREM\_ADD
- ^B001 - IESP\_A\_D
- ^B000 - IRMC\_A\_D

### 3.8.3 MIF DMUX

This mux controls the data source to the memories. Encoding is as follows:

- ^B11 - Illegal
- ^B10 - IREM\_DAT
- ^B01 - IESP\_A\_D
- ^B00 - IRMC\_A\_D

### 3.8.4 MIF RAS/CAS MUX

This mux controls the address for the DRAM. RCMUX\_SEL is bit 0 and MEM\_SIZE is bit 1 of the mux control. Encoding is as follows:

- RCMUX\_SEL = 0, select RAS
- RCMUX\_SEL = 1, select CAS
  
- MEM\_SIZE = 0, select 9 bit addressing
- MEM\_SIZE = 1, select 10 bit addressing
  
- ^B11 - DRAM\_ADD<9:0> = CAS\_CNT<9:0>
- ^B10 - DRAM\_ADD<9:0> = SELAD<19:10>
- ^B01 - DRAM\_ADD<9:0> = CAS\_CNT<9:0>
- ^B00 - DRAM\_ADD<9:0> = SELAD<18:9>

### 3.8.5 CAS COUNTER

The CAS counter is a 10-bit counter which loads on LCAS and increments on ICAS. The input is SELAD<9:0> and the output is CAS\_CNT<9:0>. If the CAS counter overflows, this is an error condition.

### 3.8.6 Misc MCSM bits

Following are the output bits of the MCSM not explicitly covered anywhere else:

OE\_PTE - A control signal allowing the output enabling from the PMC of bits 31 thru 16.

OE\_FV - A control signal allowing the output enabling from the PMC of bits 15 thru 0.

OE\_OWN - A control signal allowing the output enabling from the PMC of bits 35 thru 32.

ENABO - A control signal allowing the BOFF REG (Byte Offset Register) to be loaded

ENAPAGE - A control signal allowing the PAGE REG to be loaded.

FORCE\_BD - Force buffer descriptor (from bit 25 of the command/address cycle) forces the SRAM address to the space assigned for the Buffer Descriptors.

### 3.8.7 SRAM CALCULATE ADDRESS functional block

The SRAM address is calculated in the following manner:

```
BEGIN
  IF FORCE_BD THEN BEGIN
    IF MEM_SIZE THEN ADDOUT<15:0> = ^H8000 + PAGE<14:0>
    ELSE ADDOUT<15:0> = ^H3800 + PAGE<10:0>;
  END;
  ELSE ADDOUT<15:0> = ^H0000 + PAGE<14:0>;
END;
```

### 3.9 THE REFRESH FUNCTIONAL BLOCK

The Refresh functional block is based on two counters run off the 80nSec clock. One counter takes care of the refresh interval and the other produces the page address to be refreshed.

It is assumed that a refresh request will always be serviced before the next is due (since refresh is top priority, this is given) and so no special care is taken to ensure that all requests are serviced.



# **CHAPTER 4**

## **THE RING ENTRY MOVER**



# CHAPTER 5

## RING ENTRY MOVER

### LORD OF THE RINGS

#### 5.1 OVERVIEW

The REM's function is move PTE's pointing to received packets (ie packets in receive rings) to transmit rings and move PTE's pointing to free pages (packets after transmission) from transmit rings back to the queue of origin of the buffer. This function controls the flow of Packets between the ESP, RMC and Adapter Manager interfaces. Through-out the REM Design Specification, discussion regarding the moving of Packets by the REM, will always refer to the moving of PTEs in the Page Table.

#### 5.2 THE PAGE TABLE

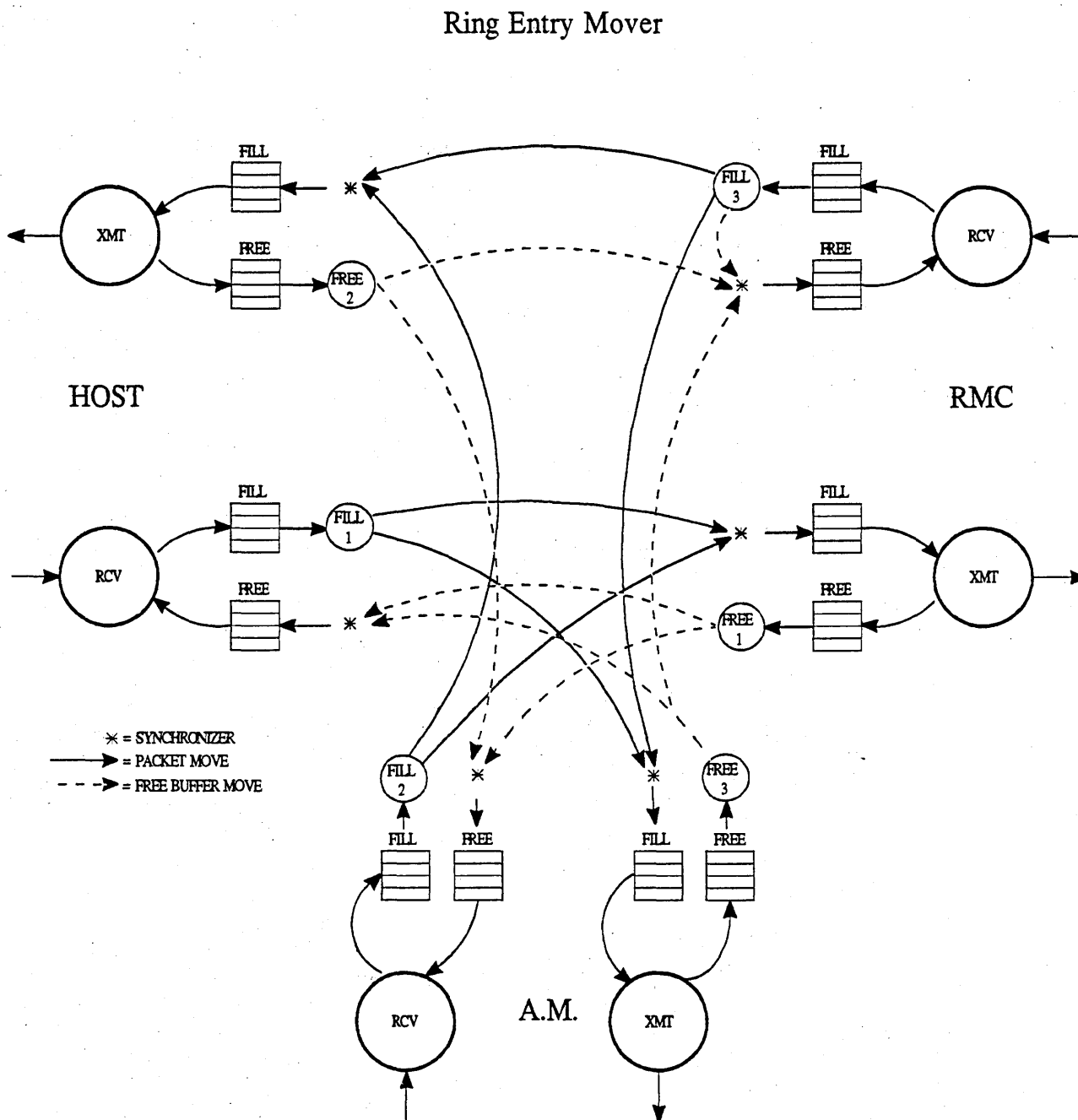
The Page Table is located in the low 14K of SRAM (in the current implementation). The Page Table is divided up into six segments or rings (as well as a number of addresses which are reserved for other uses). Of the rings, there is a transmit and receive ring for each of the three (HOST, AM and RMC) interfaces to the PMC. The rings are modeled after the RMC rings. They are listed as follows:

- RMC\_RCV ring (Ring Memory Controller receive ring)
- RMC\_XMT ring (Ring Memory Controller transmit ring)
- AM\_RCV ring (Adapter Manager receive ring)
- AM\_XMT ring (Adapter Manager transmit ring)
- HOST\_RCV ring (HOST receive ring)
- HOST\_XMT ring (HOST transmit ring)

##### 5.2.1 LEGAL PATHS FOR PTEs BETWEEN PAGE TABLE RINGS

- RMC\_RCV ring to AM\_XMT ring
- RMC\_RCV ring to HOST\_XMT ring
- HOST\_RCV ring to RMC\_XMT ring
- HOST\_RCV ring to AM\_XMT ring
- AM\_RCV ring to HOST\_XMT ring
- AM\_RCV ring to RMC\_XMT ring

Figure 5-1: PERMITTED 'MOVES' OF THE RING ENTRY MOVER



XPA\_HYB\_OPT\_REM.UIS

## 5.2.2 FILL AND FREE SPACE AT INITIALIZATION

Each of the receive rings in the Page Table is broken up into FILL space and FREE space. FILL space is initialized with a series of Page Table Entries each having a unique Physical Page Number (PPN). The sum of all PPNs in FILL space in the three receive rings must be less than the number of pages or buffers in the Packet Buffer Memory. This is because the firmware requires some number of buffers for CNS, some private allocation and possibly some other purposes. The color and own\_bit fields are also initialized for PTEs in FILL space. FREE space is initialized with PTEs that do not have an assigned Physical Page Number, hence FREE space is also referred to as unassigned space. The only valid field at initialization in receive ring FREE space, is the own bit.

Each of the transmit rings in the Page Table also has FILL and FREE space. However, at initialization, the base of FILL space and FREE space refer to the same base address and area, in the ring. The only valid field of the PTEs in the transmit ring at initialization is the own\_bit.

Firmware initializes all unused PTE's with values having even parity. Since the system is normally operated in odd parity mode an accidental access to an unused PTE will result in a MIF parity error thereby stopping the REM from further activity.

Figure 5-2: THE SIX RINGS AT INITIALIZATON

The PTEs that compose the XMT and RCV RINGS for the AM,HOST and RMC, have been substituted with either PPN#(Physical Page Number) or UNAS (Unassigned Space). PPNs are allocated to the segments of the RING which actually have valid Physical Page Numbers. UNAS is designated to the segments of the RING which do not have Valid Physical Page Numbers.

The OWN bit is CLEAR when the REM has ownership of the RING and it is SET when the external interface has Ownership.

Pointers move in an upward direction.

The external interface pointers are preceded by EXT.

| RMC_RCV_RING       | AM_RCV_RING       | HOST_RCV_RING        |
|--------------------|-------------------|----------------------|
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0 <--RMC_RCV | UNAS  0 <--AM_RCV | UNAS  0 <--HOST_RCV  |
| +-----+            | +-----+           | +-----+              |
| FREE               | FREE              | FREE                 |
| +-----+            | +-----+           | +-----+              |
| PPN5  1            | PPN10  1          | PPN15  1             |
| +-----+            | +-----+           | +-----+              |
| PPN4  1            | PPN9  1           | PPN14  1             |
| +-----+            | +-----+           | +-----+              |
| PPN3  1            | PPN8  1           | PPN13  1             |
| +-----+            | +-----+           | +-----+              |
| PPN2  1            | PPN7  1           | PPN12  1             |
| +-----+            | +-----+           | +-----+              |
| PPN1  1 <--RMC_RCV | PPN6  1 <--AM_RCV | PPN11  1 <--HOST_RCV |
| +-----+            | +-----+           | +-----+              |
| FILL               | FILL              | FILL                 |
| EXT_RMC            | EXT_AM            | EXT_HOST             |
| PNTR               | PNTR              | PNTR                 |
|                    |                   |                      |
| RMC_XMT_RING       | AM_XMT_RING       | HOST_XMT_RING        |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0            | UNAS  0           | UNAS  0              |
| +-----+            | +-----+           | +-----+              |
| UNAS  0 <--RMC_XMT | UNAS  0 <--AM_XMT | UNAS  0 <--HOST_XMT  |
| +-----+            | +-----+           | +-----+              |
| FILL               | FILL              | FILL                 |
| RMC_XMT            | AM_XMT            | HOST_XMT             |
| FREE               | FREE              | FREE                 |
| EXT_RMC            | EXT_RMC           | EXT_HOST             |
| PNTR               | PNTR              | PNTR                 |

rings\_at\_init.line\_art

### 5.2.2.1 RING POINTERS

The REM has twelve registers initialized with the base address of either FREE or FILL space for each ring. The registers are listed in TABLE\_1 with a description of the addresses that the Adapter Manager will initialize them with.

**Table 5-1: REM REGISTERS WITH RING POINTERS**

| REGISTER      | ADDRESS AT INITIALIZATION                             |
|---------------|-------------------------------------------------------|
| RMC_RCV_FIL   | Base address of the RMC_RCV ring                      |
| RMC_RCV_FREE  | Base address of unassigned space in the RMC_RCV ring  |
| RMC_RCV_SIZE  | Address of last page in the RMC_RCV ring              |
| RMC_XMT_FIL   | Base address of the RMC_XMT ring                      |
| RMC_XMT_FREE  | Base address of the RMC_XMT ring                      |
| RMC_XMT_SIZE  | Address of last page in the RMC_XMT ring              |
| AM_RCV_FIL    | Base address of the AM_RCV ring                       |
| AM_RCV_FREE   | Base address of unassigned space in the AM_RCV ring   |
| AM_RCV_SIZE   | Address of last page in the AM_RCV ring               |
| AM_XMT_FIL    | Base address of the AM_XMT ring                       |
| AM_XMT_FREE   | Base address of the AM_XMT ring                       |
| AM_XMT_SIZE   | Address of last page in the AM_XMT ring               |
| HOST_RCV_FIL  | Base address of the HOST_RCV ring                     |
| HOST_RCV_FREE | Base address of unassigned space in the HOST_RCV ring |
| HOST_RCV_SIZE | Address of last page in the HOST_RCV ring             |
| HOST_XMT_FIL  | Base address of the HOST_XMT ring                     |
| HOST_XMT_FREE | Base address of the HOST_XMT ring                     |
| HOST_XMT_SIZE | Address of last page in the HOST_XMT ring             |

Note: Number of pages in a ring must be a power of 2

### 5.2.3 PAGE TABLE ENTRIES

The Page Table Entries (PTEs) in the receive rings are comprised of the following:

- Forwarding Vector with parity <15:0>
- Physical Page Number in the Packet Buffer with parity <31:16>
- Own Bit (Clear for REM ownership and set for external interface ownership)<34>



**Table 5-2: PTE ASSIGNMENTS FOR FILL AND FREE SPACE AT INITIALIZATION**

| RING SEGMENT  | OWN_BIT | COLOR | PPN | FV |
|---------------|---------|-------|-----|----|
| RMC_RCV_FIL   | SET     | RMC   | YES | -  |
| RMC_RCV_FREE  | CLEAR   | -     | -   | -  |
| RMC_XMT_FIL   | CLEAR   | -     | -   | -  |
| RMC_XMT_FREE  | CLEAR   | -     | -   | -  |
| AM_RCV_FIL    | SET     | AM    | YES | -  |
| AM_RCV_FREE   | CLEAR   | -     | -   | -  |
| AM_XMT_FIL    | CLEAR   | -     | -   | -  |
| AM_XMT_FREE   | CLEAR   | -     | -   | -  |
| HOST_RCV_FIL  | SET     | HOST  | YES | -  |
| HOST_RCV_FREE | CLEAR   | -     | -   | -  |
| HOST_XMT_FIL  | CLEAR   | -     | -   | -  |
| HOST_XMT_FREE | CLEAR   | -     | -   | -  |

### 5.3 SETTING THE STAGE FOR REM ACTIVITY

Packets will be written into the Packet Buffer Memory via the PMC when the SOURCE (SRC) of the Packet, either the Host, Adapter Manager or the RMC, sends the appropriate commands to the PMC to initiate the process. The SRC interface has a local register which is initialized with the base address of the FILL section of the SRC\_RCV ring in the Page Table. The source will send this address and the byte offset of the Packet with the command to write data into the Packet Buffer Memory. The PMC will go to this address in the Page Table and read the PPN field to determine where in the Packet Buffer the first page of the Packet should be written. The SRC will monitor the number of bytes it has written into the PBM and will increment the base address in its local register when the last byte for that page has been written. The SRC will then send the incremented address with the next write command. The PMC will now read the PPN from the PTE specified by the address, and the page now indicated will be written. At the time that the Packet has been completely written into the Packet Buffer, the own bit will have been cleared for each PTE that was accessed during the write of the Packet. The OWN bit of the first PTE accessed, will be cleared last. The SRC of the Packet will next send either RMC\_RCV\_DONE, AM\_RCV\_DONE, or HOST\_RCV\_DONE, to the REM.

The explanation above of the exchange between the external interface and the PMC is only intended to provide a partial overview of the actual process to provide to the reader a better understanding of where the REM fits into the scheme. The importance of Buffer Descriptors and the mechanism for setting the Own bits have not been discussed. For a complete understanding of this interface, the reader should refer to the Memory Interface (MIF) portion of the Design Specification.

SRC\_RCV\_DONE 'wakes-up' the REM MOVE\_FILL State Machine. If the RCV\_DONE was received from the Host or the Adapter Manager, the REM will execute without buffer management. However, packets received from the RMC are monitored and under certain circumstances may not be delivered because Buffer Management is included in this path.

## **5.4 BUFFER MANAGEMENT IN THE PACKET MEMORY CONTROLLER**

The REM will manage the storage of Packets in the PMC by manipulating PTEs.

### **5.4.1 Buffer Management for the HOST Interface**

The REM will never discard Packets written to the PMC from the Host. By default, Packets from the Host will always be destined for the RMC. Only during a maintenance operation will a Packet be destined to the Adapter Manager from the Host. By default, the REM will copy the PTEs of the Host Packet from the FILL section of the HOST\_RCV ring to the FILL section of the RMC\_XMT ring and thereby transfer the Host Packet to the RMC interface. Refer to CSR00 definition for details

During a Maintenance operation, the REM will copy the PTEs in the FILL section of the HOST\_RCV ring to the FILL section of the AM\_XMT ring. The REM will be aware that a Maintenance operation is in effect by a particular CSR being SET. The CSR bit in the SET state will indicate to the REM that this Packet is intended for the Adapter Manager.

### **5.4.2 Buffer Management for the ADAPTER MANAGER Interface**

The REM will never discard Packets written into the PMC from the Adapter Manager. The REM will read the HOST bit in the Forwarding Vector to determine if the Packet is intended for the Host or the RMC. If the Host bit is SET it indicates that this Packet is intended for the Host. If the Host bit is CLEAR, the REM will copy the PTEs of the AM Packet from the FILL section of the AM\_RCV ring to the FILL section of the RMC\_XMT ring. If the Host bit is SET, the REM will copy the PTE(s) to the FILL section of the HOST\_XMT ring. (Note - PTE(s) representing Packets generated by the AM will have only one valid bit in the Forwarding Vector field, the Host bit.)

### **5.4.3 Buffer Management for the RMC Interface**

Packets received from the RMC will be delivered to the AM, the Host, or discarded. The REM will read the Host bit to determine which interface the Packet is going to. If the Host bit is SET the Packet will be destined to the Host. When the Host bit is CLEAR, the Packet is intended for the AM. Under certain conditions, the REM will discard a Packet otherwise intended for one of these interfaces. A Packet will be discarded if the Discard bit has been set in the Forwarding Vector or if the REM determines that there are not enough buffers available in the allocation for a specific type of Packet. If there is sufficient allocation and



the Discard bit is not SET, the REM will copy a PTE slated for the AM from the FILL section of the RMC\_RCV ring to the FILL section of the AM\_XMT ring. If the PTE is slated for the Host, the REM will copy the PTE from the FILL section of the RMC\_RCV ring to the FILL section of the HOST\_XMT ring.

#### 5.4.3.1 Buffer Allocation and RMC Packets

Packets received from the RMC will be classified as either Host Packets or Adapter Manager Packets. Adapter Manager Packets can be subdivided into four types: AM, SMT, MOP, and ERROR Packets. Therefore, there are a total of five categories of Packets received from the RMC. A mechanism has been established to divide a subset of the available buffers between the five types of Packets received from the RMC. The mechanism involves assigning an ALLOCATION to each RMC Packet-type. The ALLOCATION is the maximum number of pages or buffers in the PBM that can be written with a specific Packet-type from the RMC, at any one time. The Allocation for each Packet-type will be written into the corresponding Allocation Register by the AM at initialization.

After a Packet has been written into the PBM from the RMC, the REM will need to determine if the number of buffers required to write the Packet was less than or equal to the allocation remaining for that specific Packet-type. To determine if this is true, the REM must first identify the Packet-type of the Packet in question. Packet-type is obtained by reading the Host bit for Host Packets and the Host and Type bits for AM Packets. The REM must also derive the number of pages that were used to write the Packet to PBM. The page count is obtained from a Register called PG\_CNT\_REG. PG\_CNT\_REG is written by the PG\_CNT\_DETERMINATION State Machine. This State Machine will read the SOP/EOP fields in the PTE and determine if the Packet is one page. If the Packet is one page, PG\_CNT\_REG will be cleared. If the Packet is a multiple page Packet, a read to the Buffer Descriptor corresponding to the first Page of the Packet will be necessary. Bits <12:0> in the Buffer Descriptor will provide the byte count for the Packet in question. The REM will send the PPN with the RD\_BD\_NXLATE command type to the MIF. The PPN will provide the index into the Buffer Descriptor Table needed to locate the appropriate Buffer Descriptor. When the REM receives the page count it calculates the corresponding number of pages - 1 and writes the result to the PG\_CNT\_REG (0 = 1\_page packet).

If the Packet is a Host Packet, the REM will read the Host Allocation Register and compare the value to the value in PG\_CNT\_REG. If the number of pages listed in PG\_CNT\_REG is less than (a page count of zero corresponds to one page) the remaining Host allocation, the REM will decrement the RMC\_HOST\_CNTR by the number in PG\_CNT\_REG. The REM will then copy the PTE(s) from the FILL section of the RMC\_RCV ring to the FILL section of the HOST\_XMT ring. If there is insufficient Host Allocation, the Packet must be discarded and the Host Discard Register, called HOST\_DISCARD\_CNTR, will be incremented by the number of buffers indicated in the PG\_CNT\_REG.

If the Packet is an AM destined Packet, the REM must read the Type field in the Forwarding Vector to determine whether it is an SMT, MOP or ERROR Packet. If the Packet is an AM Packet and it is not either a SMT, MOP, or ERROR type Packet, the REM will read the AM Allocation Register called the RMC\_AM\_CNTR and compare this value with the value in PG\_CNT\_REG. If the value in the PG\_CNT\_REG is less than the value in RMC\_AM\_CNTR, the REM will decrement RMC\_AM\_CNTR by the number of buffers indicated in PG\_CNT\_REG. The REM will then copy the PTE(s) from the FILL section of the RMC\_RCV

ring to the FILL section of the AM\_XMT ring. If there is not adequate AM allocation the Packet will be discarded and the AM\_DISCARD\_CNTR will be incremented by the value in PG\_CNT\_REG.

If the Packet is an AM Packet as indicated by the Host bit, and the Packet-type is either SMT or MOP, the REM will check to see whether there is adequate allocation remaining in the RMC\_AM\_CNTR. If there is, the REM will decrement RMC\_AM\_CNTR by the value in PG\_CNT\_REG. The REM will then copy the PTE(s) from the FILL section of the RMC\_RCV ring to the FILL section of the AM\_XMT ring. If there is not enough allocation, the REM will check the allocation in either the SMT or MOP Allocation Registers. These registers have been allotted with a certain number of buffers which are additional to the total assigned to AM Packets. Hence, they are referred to as 'overdraft' allocations. The REM will compare the value in PG\_CNT\_REG with the value in the SMT or MOP overdraft register called RMC\_SMT\_OD\_CNTR or RMC\_MOP\_OD\_CNTR, respectively. If the value in PG\_CNT\_REG is less than in the specified overdraft register, the REM will decrement the overdraft register by the number of buffers in PG\_CNT\_REG. The REM will then copy the PTE(s) from the FILL section of the RMC\_RCV ring to the FILL section of the AM\_XMT ring. If the allocation is less than what is required, the Packet is discarded and either the SMT\_DISCARD\_CNTR or the MOP\_DISCARD\_CNTR is incremented by the value in PG\_CNT\_REG.

If the Packet is an AM Packet and the Type field indicates an ERROR Packet the REM will check to see whether there is adequate allocation remaining in the RMC\_ERROR\_CNTR. If there is not, the Packet will be discarded and the ERROR\_DISCARD\_CNTR will be incremented by the page count. If there is, the REM must now check to see if there is also enough AM Allocation remaining in the RMC\_AM\_CNTR Register. If there is sufficient allocation, the REM will decrement the RMC\_ERROR\_CNTR by the value in PG\_CNT\_REG + 1. The REM will also decrement the RMC\_AM\_CNTR by the value in PG\_CNT\_REG + 1. The REM will copy the PTE(s) from the FILL section of the RMC\_RCV ring to the FILL section of the AM\_XMT ring. If either allocation is less than what is required, the Packet is discarded and the ERROR\_DISCARD\_CNTR will be incremented by the value in PG\_CNT\_REG + 1, but the AM\_DISCARD\_COUNTER will *NOT* be incremented.

The buffer count in the RMC\_ERROR\_CNTR is a subset of the total number of buffers in the AM Allocation and represents how many buffers may be used for the delivery of Error Packets. It will be a programmable number. If the remaining Allocation in the RMC\_ERROR\_CNTR is less than the needed number of buffers, the Packet will be discarded. Hence, though there may have been enough AM Allocation, if the ERROR Allocation has been exhausted the Packet will be discarded.

#### 5.4.3.2 Discarding a Packet

Only Packets received from the RMC interface can be discarded. An RMC Packet will be discarded in two instances, either the Parser-Discard bit will have been set in the Forwarding Vector or secondly, there is insufficient Allocation remaining for the Packet-type at hand.

A Packet that has been written into the Packet Buffer Memory from the RMC interface is represented in the Page Table by a number of PTE(s) equal to the number of pages that were written. If the Packet must be discarded, the PTE(s) need to be recycled to allow them to be used for future incoming RMC Packets. The recycling or Discard procedure involves writing the PTE(s) representing the Packet from the FILL section of the RMC\_RCV ring to the FREE section of the RMC\_RCV ring and setting the ownership bit(s). With each PTE written to the FREE section of the RMC\_RCV ring, the FREE pointer is incremented. Hence, the PTE(s) are now part of the FILL section of the RMC\_RCV ring once again, thereby returning them to the pool of available buffers.

The REM maintains statistics on discarded Packets by incrementing the appropriate Discard Register by the number of buffers used for the Discarded Packet. The buffer count will be determined from the PG\_CNT\_DETERMINATION State Machine. The buffer count will be stored in the PG\_CNT\_REG Register.

If the Discard bit is SET in the Forwarding Vector, the Packet is Discarded as described above and the PARSER\_DISCARD Register is incremented by the number of buffers listed in PG\_CNT\_REG.

If the Packet is being Discarded due to insufficient Allocation, the Packet is Discarded as described above and the appropriate Discard Register will be incremented. The REM must determine which DISCARD Register needs to be incremented in this case. Therefore, the REM must first read the Host bit to determine whether the Packet is an AM or Host type. If it is a Host type the HOST\_DISCARD Counter will be incremented by the buffer count in PG\_CNT\_REG. If the Host bit indicates an AM Packet, then the Type field must be read to determine if it is an AM, SMT, MOP or ERROR Packet. If it is an AM Packet the AM\_DISCARD Counter is incremented. If it is a SMT or MOP Packet the SMT\_DISCARD Counter or MOP\_DISCARD Counter is incremented. If it is an ERROR Packet the ERROR\_DISCARD Counter is incremented.

Refer to the Table below for a listing of the Allocation and Discard Registers.

**Table 5-3: REM PACKET\_TYPE ALLOCATION REGISTERS AND DISCARD REGISTERS**

| PACKET-TYPE ALLOCATION | DISCARD             |
|------------------------|---------------------|
| RMC_HOST_CNTR          | HOST_DISCARD_CNTR   |
| RMC_AM_CNTR            | AM_DISCARD_CNTR     |
|                        | PARSER_DISCARD_CNTR |
| RMC_SMT_OD_CNTR        | SMT_DISCARD_CNTR    |
| RMC_MOP_OD_CNTR        | MOP_DISCARD_CNTR    |
| RMC_ERROR_CNTR         | ERROR_DISCARD_CNTR  |

### 5.4.3.3 Buffer Allocation and Ring Configuration

The Page Table will be comprised of 14K, thirty-two bit addresses or entries at FRS. It will be expandable to 32K pages to support 4M of DRAM Memory. The bulk of the entries in the Page Table will be divided into six rings, three Transmit and three Receive rings. The remaining entries will be reserved for other uses. Each of the Receive rings must always have at least one unassigned buffer. The Transmit rings must always have a number of addresses larger than the total number of buffers which can fill it.

The RMC\_RCV\_RING will have its buffer assignments divided between the following allocations:

- Host Allocation
- Adapter Manager Allocation
  - Error Allocation is included but is a programmable number
  - SMT is included
  - MOP is included
  - AM
- SMT\_OD Allocation
- MOP\_OD Allocation
- PLUS at least 12 unassigned buffers

The total number of RMC\_RCV ring buffers is less than or equal to the (RMC\_RCV ring size - 1)

#### 5.4.3.3.1 NOTE ON HOST BUFFER COUNT

The REM will monitor the number of buffers remaining in the FILL section of the HOST\_RCV ring. If the number of buffers falls to eighteen or less, the REM will deassert the signal to the ESP interface called PBI\_AV. There will be a Register called HOST\_RCV\_COUNT which will act as the counter. At initialization, the Register will have the total number of buffers that can be written in Packet Buffer Memory by the HOST. Buffers are claimed in the HOST\_RCV ring by the process of writing a Packet into the Packet Buffer Memory. The Counter will be decremented by the number of PTEs used for the Packet. As buffers are recycled back to the HOST\_RCV ring by the MOVE\_FREE State Machines, the Register will be incremented.

## 5.5 NOTIFYING THE INTERFACE AND WAKING THE MOVE\_FREE STATE MACHINE

Buffer Management and Discarding a Packet are Component State Machines in the MOVE\_FILL process. The procedure which copies the PTEs from the RMC\_RCV ring to the DESTINATION\_XMT ring is also a component State Machine of the MOVE\_FILL process. The object of the MOVE\_FILL process is to copy the PTEs to the Destination ring. Once the steps of the copy have been completed, the REM will send the DESTINATION\_START\_XMT command to the corresponding interface.

START\_XMT notifies the destination interface that there is a Packet to be read from the Packet Buffer Memory. The destination interface has a local register containing the base address of the FILL section of the DESTINATION\_XMT ring. The destination will send a RD\_BD\_DATA command with this address and a byte offset to the PMC. The MIF will do a translation to locate the BD in the Buffer Descriptor Table (BDT). The destination interface will then send a RD\_DATA\_REQ command to the PMC. The base address of the FILL segment of the DESTINATION\_XMT ring and a byte-offset will be sent with the command to the PMC. The MIF will do a translation to determine the Physical address of the first page of the Packet in the Packet Buffer Memory. The Destination will begin reading the first 4 or 8 longwords of the first page of the Packet. Every 4 or 8 longwords, the destination will send RD\_DATA\_REQ. At the end of the page, the destination may send CLR\_OWN\_PTE to the PMC. The MIF will then clear the own bit for the PTE which returns ownership to the REM. The process is now repeated for the next page until the end of Packet (EOP) is reached. At the point when the entire Packet has been read from the PBM, the destination will send DESTINATION\_XMT\_DONE to the REM. The REM now executes the procedure in the MOVE\_FREE State Machine which recycles PTE(s) to the RCV ring of origin.

## 5.6 RING CONFIGURATIONS

Please refer to the Appendix for several illustrations of ring configurations. They will lend a pictorial understanding of the MOVE\_FILL and MOVE\_FREE State Machines as well as the DISCARD State Machine.

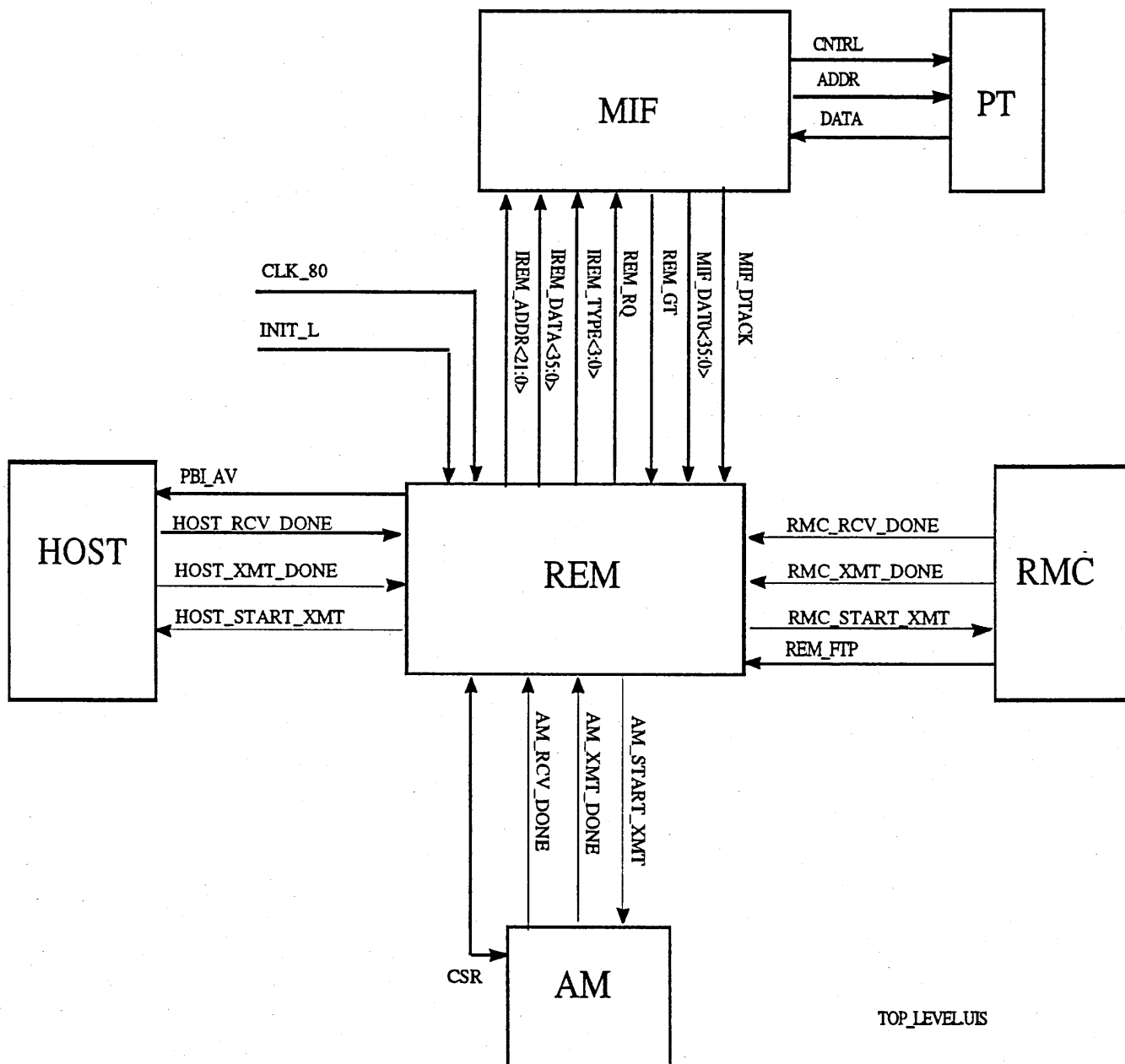
## 5.7 REM FUNCTIONAL BLOCKS

The REM CONTEXT drawing on the following page illustrates the REM's relation to the MIF portion of the PMC gate array as well as to the ESP, RMC and Adapter Manager interfaces and to the Page Table. The PMC gate array includes only the REM and MIF blocks shown here.

The illustration of the Expansion of the REM shows the three top level state machines and the CSR control logic block as well as the shared Registers. The MOVE\_FREE, MOVE\_FILL, and Arbitration state machines can be further expanded to their component state machines.

Figure 5-4: TOP LEVEL BLOCK DIAGRAM OF THE RING ENTRY MOVER

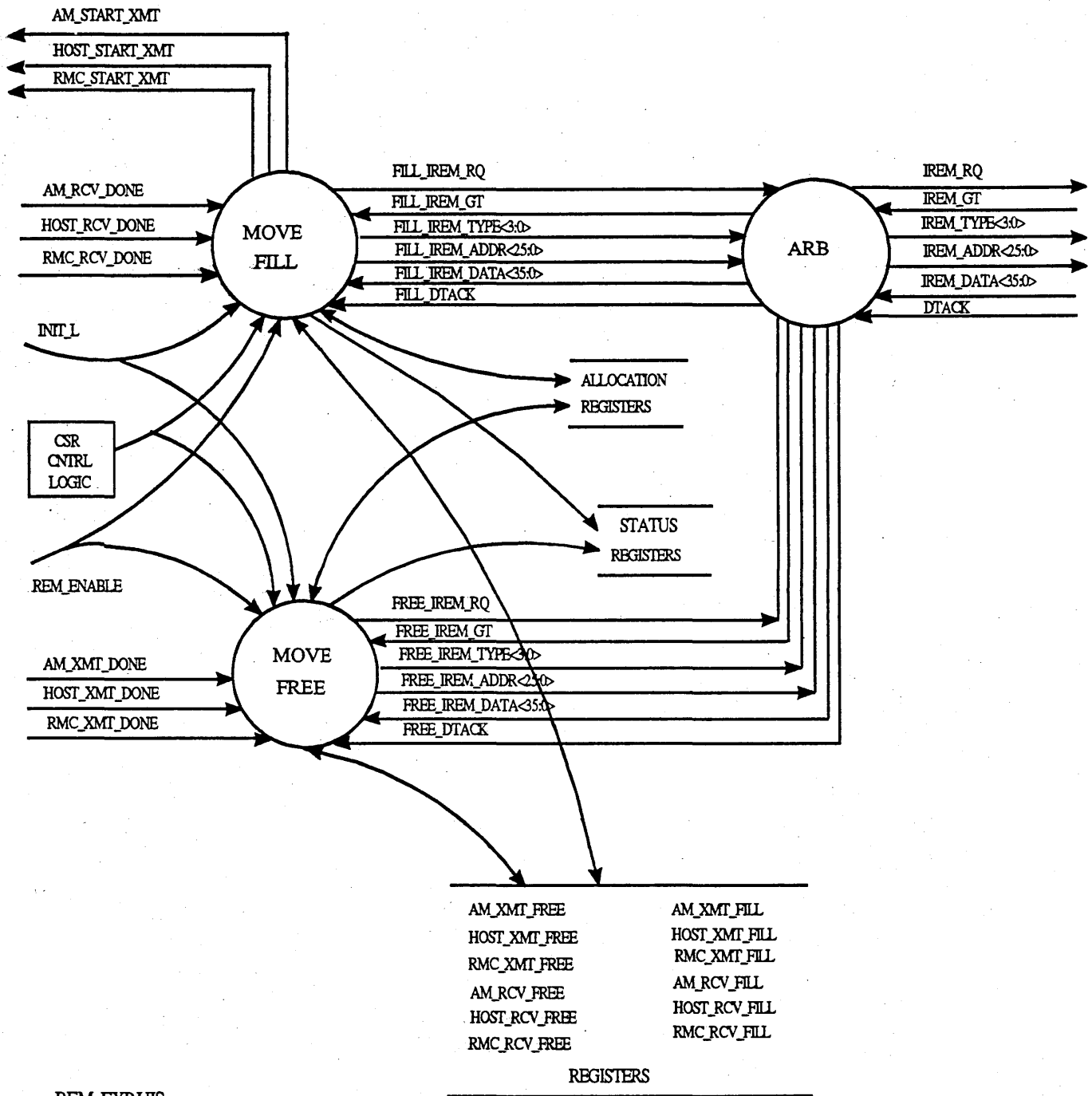
REM CONTEXT DRAWING



5.7.1 EXPANSION OF THE REM

Figure 5-5: EXPANSION OF THE REM BLOCK

EXPANSION of the REM



REM\_EXP.UIS

MAY 9,1989



## 5.8 EXPANSION OF THE RING ENTRY MOVER

### 5.8.1 MOVE\_FILL BLOCK

#### 5.8.1.1 MOVE\_FILL SUPERVISORY STATE MACHINE FLOW

There are three interfaces to the REM. Each interface will be writing Packets into the Packet Buffer Memory. Each time a Packet is written, the REM will be required to move the PTEs corresponding to the Packet, from the SRC\_RCV ring to the DEST\_XMT ring. This process is handled by the MOVE\_FILL state machine. The Supervisory state machine will determine which source ring will be serviced in which order. The MOVE\_FILL Supervisory state machine will transition from the IDLE state when INT\_SRC\_RCV\_DONE is true from any of the three interfaces. The Supervisory state machine flow will generally proceed from RMC to Host to AM to RMC again and if the INT\_SRC\_RCV\_DONE is asserted it will assert the appropriate SRC\_FILL\_GRANT which will activate the MOVE\_FILL state machine. There are a few interesting details which will affect this flow that need to be addressed. They are listed below:

- If FTP\_REM is asserted, it indicates that there is a transmit in progress from the RMC\_XMT ring. When FTP\_REM is asserted, emphasis should be placed on attempting to fill the RMC\_XMT ring with new PTEs corresponding to Packets to be transmitted. The REM will strive to keep a loaded RMC\_XMT ring to take full advantage of the Token when it is received. If FTP\_REM is asserted, priority is given to moving packets from the HOST\_RCV and AM\_RCV rings (in that order) to the RMC\_XMT ring.
- If FTP\_REM is true but neither INT\_HOST\_RCV\_DONE or INT\_AM\_RCV\_DONE are asserted, then the Supervisory state machine issues RMC\_FILL\_GRANT and moves from the RMC\_RCV ring are allowed
- If INT\_HOST\_RCV\_DONE is asserted, the Supervisory state machine issues HOST\_FILL\_GRANT and the HOST\_RCV ring is serviced. When PKT\_MOVE\_COMPLETE has been set by the MOVE\_FILL logic, the SSM will check to see if FTP\_REM is asserted. If FTP\_REM is asserted and INT\_HOST\_RCV\_DONE is set, the SSM will assert HOST\_FILL\_GRANT once again and service the HOST\_RCV ring.
- In order to help prevent the RMC receive FIFO from backing up with potential for overflow, the Supervisory State Machine gives priority to servicing the RMC\_RCV ring (unless FTP\_REM is asserted). The state machine maintains a variable RMC Loop Count (RLC) which is initialized to 255. Each time the SSM grants the RMC *SUCCESSIVELY* the variable is decremented. In order to prevent lockout, should the variable reach the value 0, then the SSM exits the loop, resets RLC to 255 and goes to service the Host or AM receive rings (should they require it).



### 5.8.1.2 OVERVIEW OF THE MOVE\_FILL STATE MACHINE

The MOVE\_FILL State Machine's object is to copy PTE(s) representing Packets that have been written into the Packet Buffer Memory from the section of the Page Table in the Source's domain to the section of the Page Table which is in the Destination's domain. Since the Destination interface has a Register that points to a subset of addresses in the Page Table, it will only become aware of a Packet if the PTE(s) are written into it's area (RCV and XMT rings) of the Page Table.

The MOVE\_FILL State Machine is composed of several component State Machines. Ultimately, the PTE(s) will be written from the FILL section of the SRC\_RCV ring to the FILL section of the DEST\_XMT ring. The PTE(s) of one Packet are transferred at a time. The copy begins with the second PTE. Following the copy of each PTE the Own bit is set relinquishing ownership to the external interface. The first PTE is copied last to accommodate the need to set the first PTE's OWN bit last. Once all the PTE(s) have been copied for the Packet, the REM sends START\_XMT to the interface that owns the Packet.

Each time a PTE is copied the DEST\_XMT\_FILL Register and the SRC\_RCV\_FILL Registers must be incremented. In Addition, if the source of the PTE is the RMC\_RCV ring, the corresponding Buffer Allocation must be decremented by the number of PTEs used to write the Packet into the PBM.

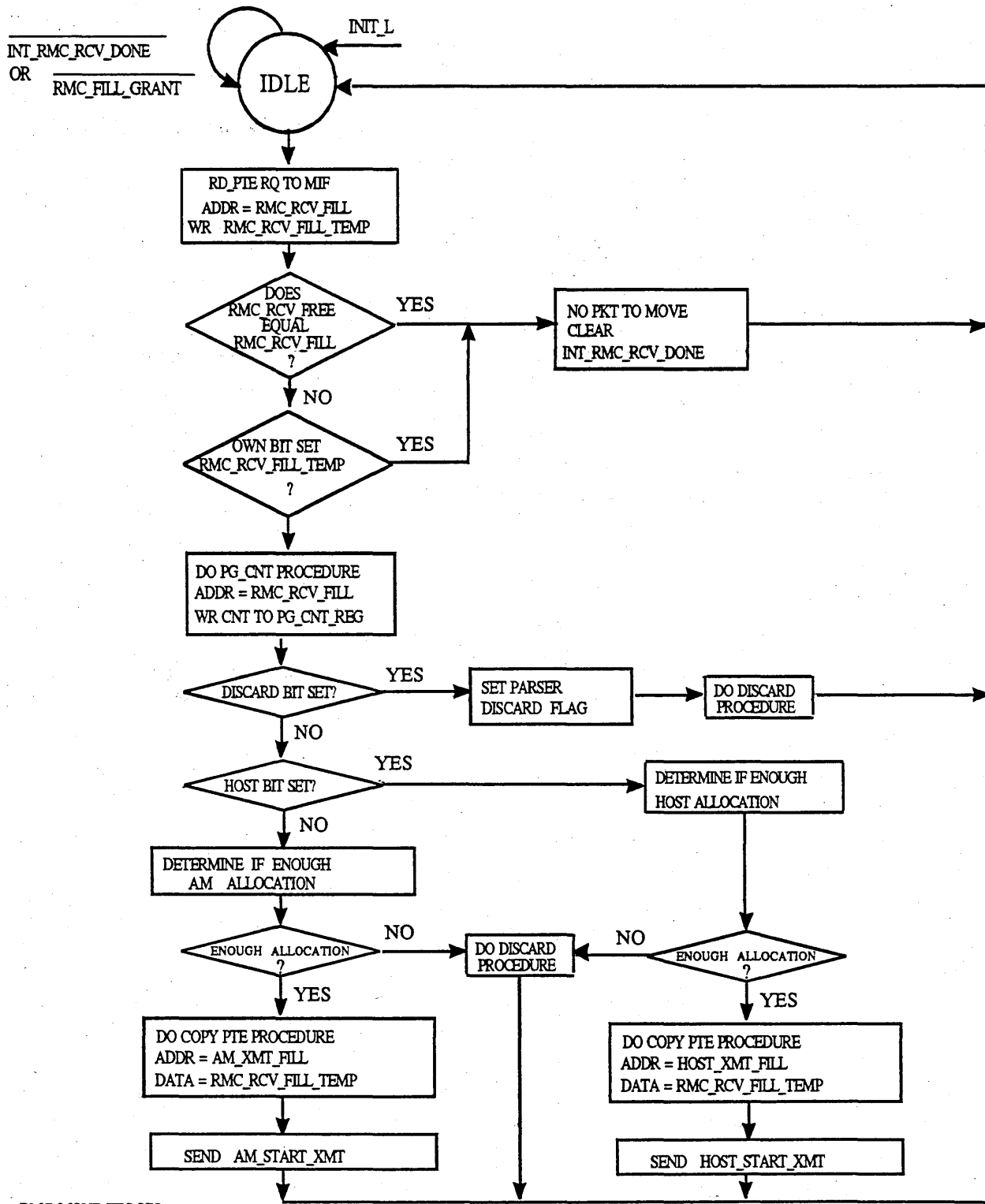
**Table 5-4: PTEs MAKE THE FOLLOWING MOVES BETWEEN RINGS DURING A MOVE\_FILL**

| RING SOURCE   | RING DESTINATION |
|---------------|------------------|
| RMC_RCV_FILL  | AM_XMT_FILL      |
| RMC_RCV_FILL  | HOST_XMT_FILL    |
| AM_RCV_FILL   | HOST_XMT_FILL    |
| AM_RCV_FILL   | RMC_XMT_FILL     |
| HOST_RCV_FILL | RMC_XMT_FILL     |
| HOST_RCV_FILL | AM_XMT_FILL      |

The following pages show the control flow of how each of the receive rings are serviced. While the implementation has only a single set of state machines, it is easier to understand the flow if each ring is shown separately.

Figure 5-7: RMC MOVE\_FILL FLOW

RMC\_MOVE\_FILL FLOW



RMC\_MOVE\_FILL.UIS

MARCH 3, 1989

Figure 5-8: HOST MOVE\_FILL FLOW

HOST\_MOVE\_FILL FLOW

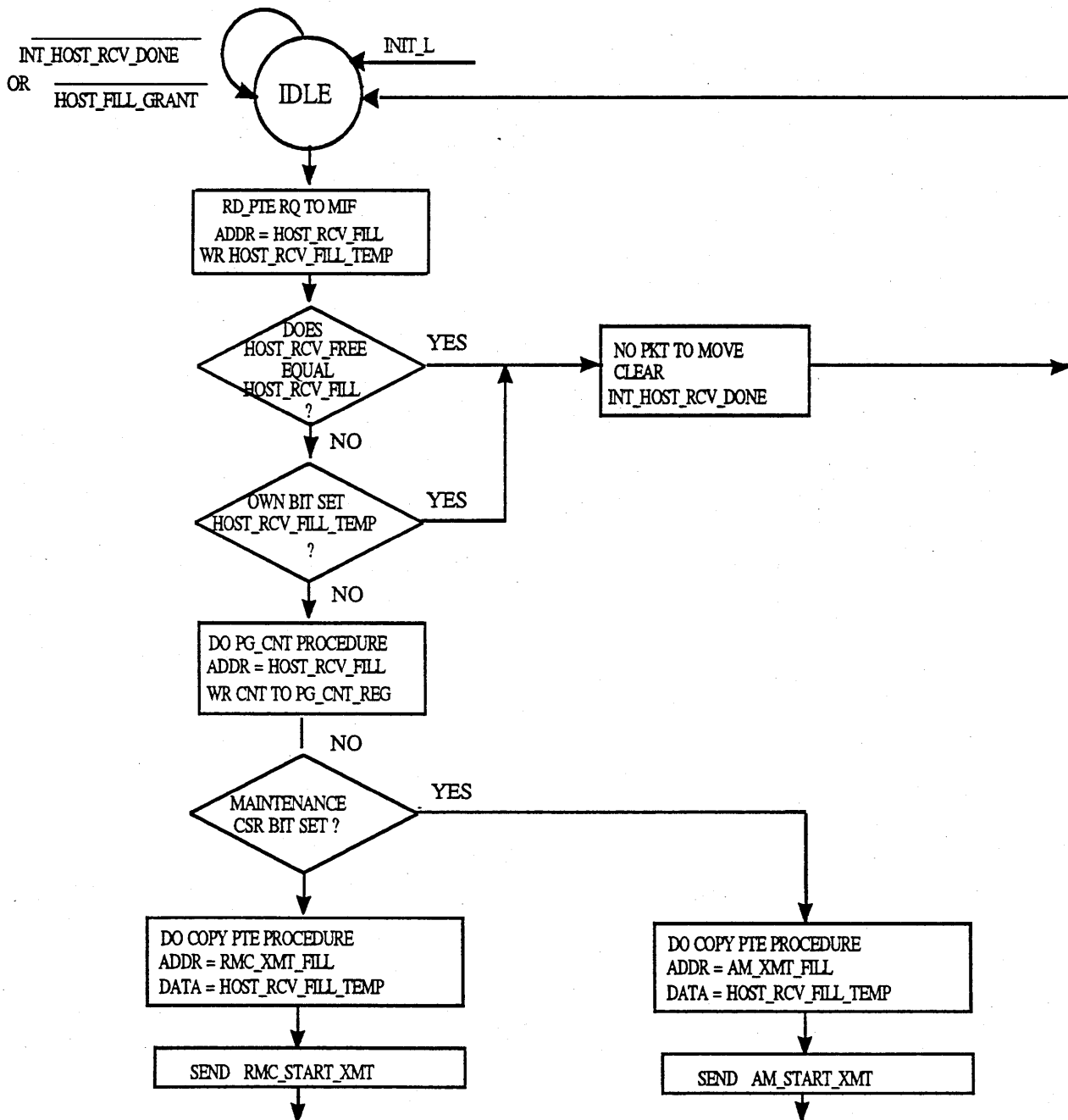
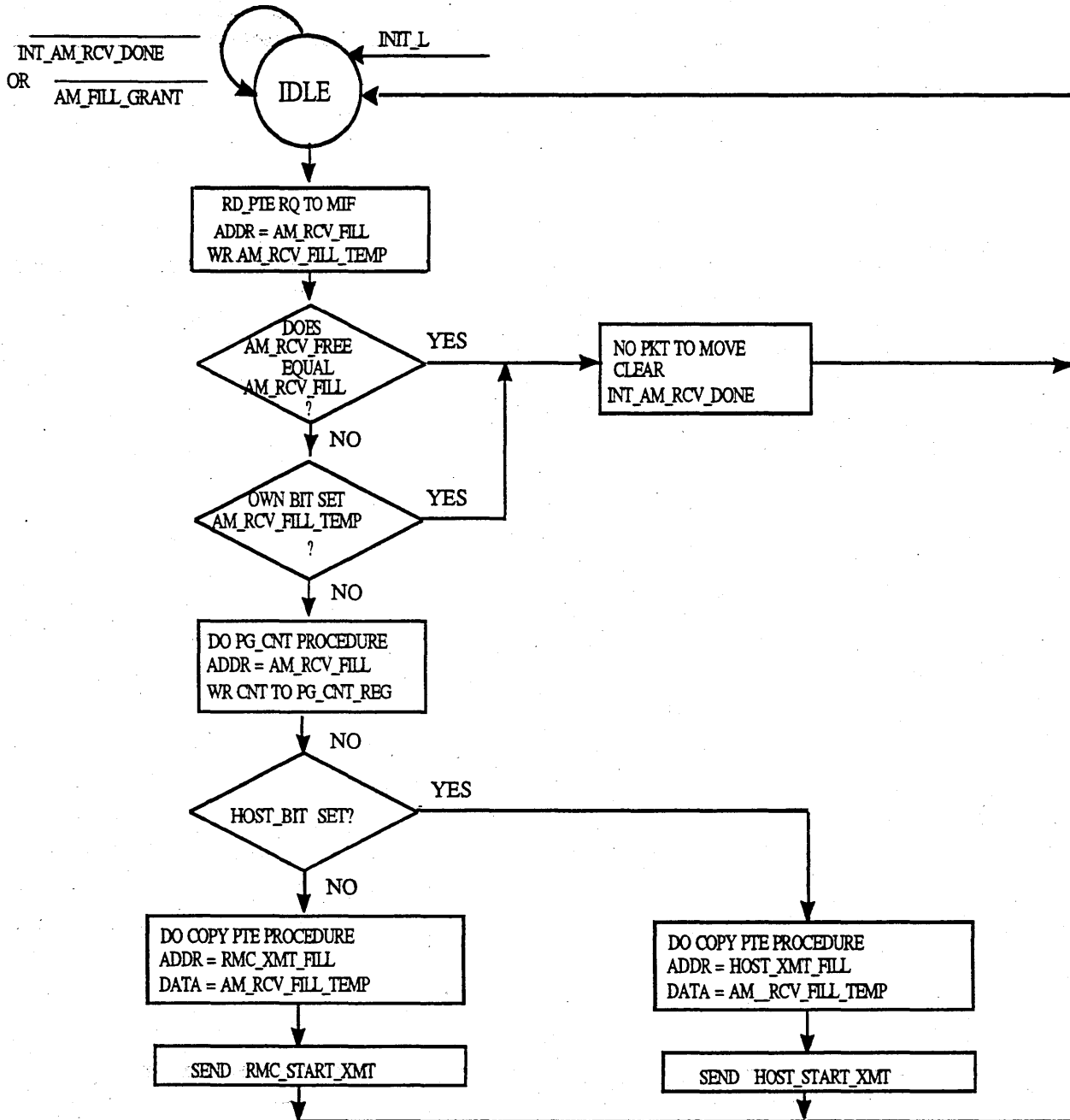


Figure 5-9: AM MOVE\_FILL FLOW

AM\_MOVE\_FILL FLOW



AM\_MOVE\_FILL.UIS  
 MARCH 3, 1989

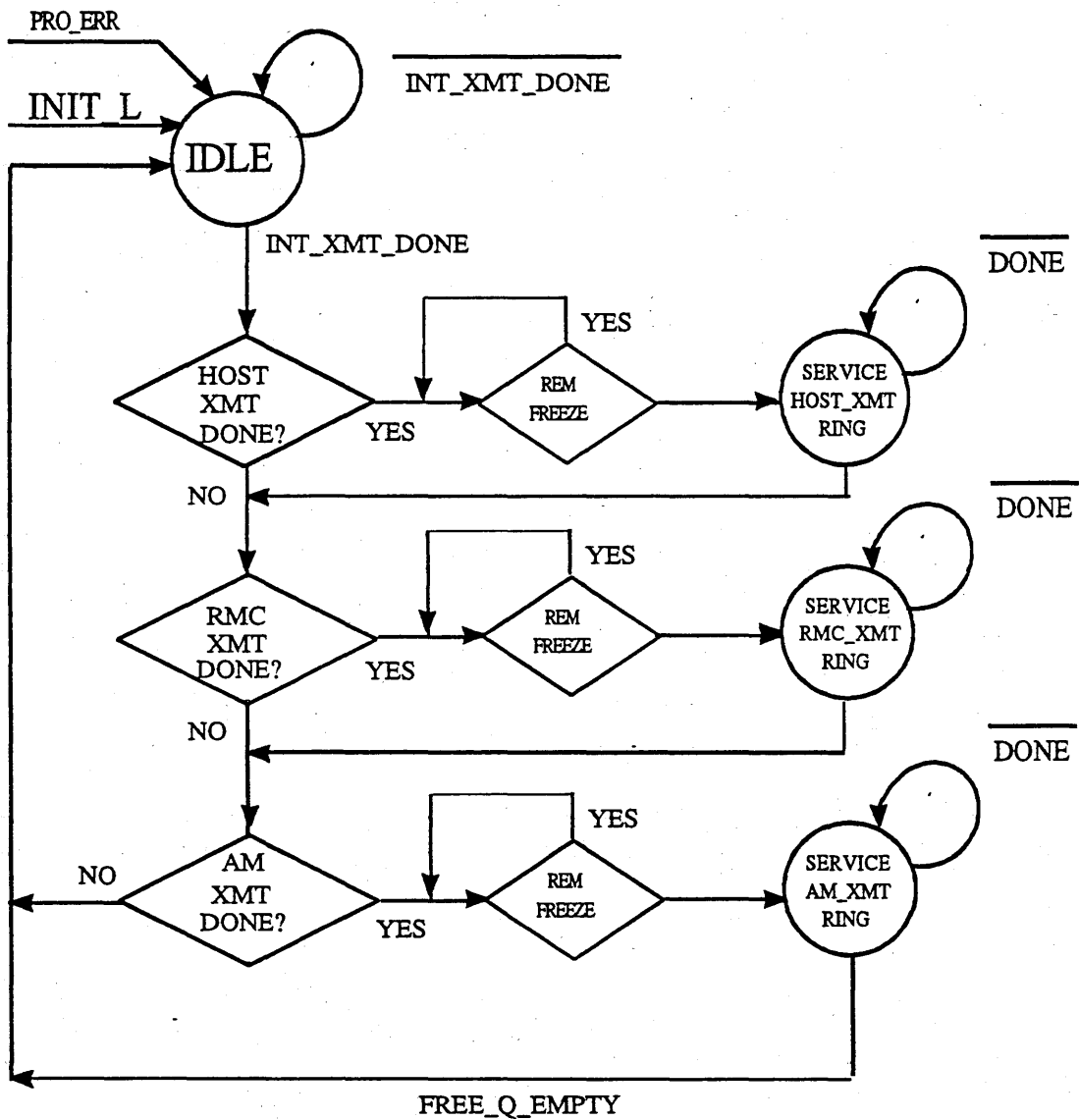
## 5.8.2 MOVE\_FREE BLOCK

### 5.8.2.1 MOVE\_FREE SUPERVISORY STATE MACHINE FLOW

There are three interfaces to the REM. Each interface will be reading Packets from the Packet Buffer Memory. Each time a Packet is read, the REM will be required to move the all the PTEs corresponding to that Packet, from the DEST\_XMT ring to the SRC\_RCV ring. This process is handled by the MOVE\_FREE state machine. The Supervisory state machine will determine which XMT ring will be serviced. The MOVE\_FREE Supervisory state machine will transition from the IDLE state when INT\_DEST\_XMT\_DONE is true from any of the three interfaces. The Supervisory state machine flow will procede from HOST to RMC to AM to HOST again and if the INT\_DEST\_XMT\_DONE is asserted, it will assert the appropriate XMT\_FREE\_GRANT thereby activating the MOVE\_FREE state machine.

Figure 5-10: MOVE\_FREE SUPERVISORY STATE MACHINE FLOW

MOVE FREE SUPERVISORY STATE MACHINE



top\_move\_free.uis



### 5.8.2.2 OVERVIEW OF THE MOVE\_FREE STATE MACHINE

The MOVE\_FREE State Machine's object is to recycle PTE(s) that had once been pointers to a Packet in Packet Buffer Memory. Once the Packet has been delivered, the PTE is returned to the ring of origin. Once the PTE has been returned, future Packets can be written into the page numbers indicated in the PPN field.

The Origin is determined by reading the Color field(s) of the PTE(s) to be recycled. The move free state machine will move up to 8 consecutive PTE's for any given ring before allowing the supervisor to proceed to the next ring. The exception to this is while servicing the AM\_XMT ring if PTE's to be returned to the RMC\_RCV ring are found. In this case the PTE's have to be returned to the RMC\_RCV ring on a packet atomic basis. This is to ensure that all allocations used are returned to the correct place as only the first PTE of a packet may contain the required information. Each PTE is moved from the FREE section of the DEST\_XMT ring to the FILL section of the RCV ring that it originated from. After each PTE is moved to the RCV ring, the Ownership bit is SET relinquishing ownership to the external interface.

Each time a PTE is written to the FREE section of the RCV ring, the DEST\_XMT\_FREE Register and the SRC\_RCV\_FREE Registers must be incremented. In Addition, if the PTE is returned to the RMC\_RCV ring, the corresponding Buffer Allocation will be increased by the number of PTEs returned.

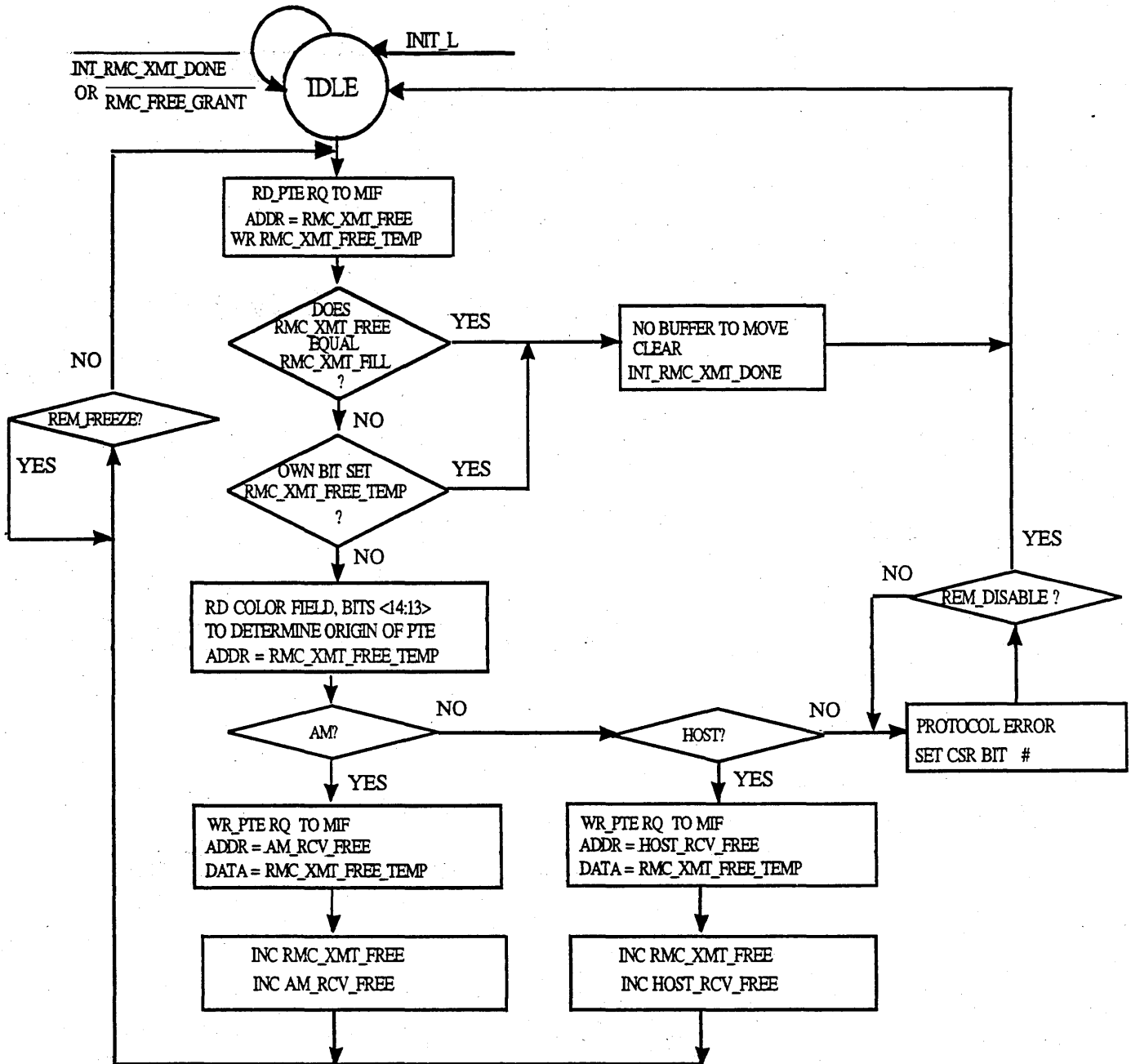
While the Move Free state machine is implemented as one state machine, for the sake of clarity the flow associated with servicing each of the XMT rings is shown.

**Table 5-5: PTEs MAKE THE FOLLOWING MOVES BETWEEN RINGS DURING THE FREE PROCESS**

| RING SOURCE   | RING DESTINATION |
|---------------|------------------|
| AM_XMT_FREE   | RMC_RCV_FREE     |
| AM_XMT_FREE   | HOST_RCV_FREE    |
| AM_XMT_FREE   | HOST_RCV_FILL    |
| AM_XMT_FREE   | RMC_RCV_FILL     |
| HOST_XMT_FREE | RMC_RCV_FILL     |
| HOST_XMT_FREE | AM_RCV_FILL      |

Figure 5-11: RMC MOVE\_FREE FLOW

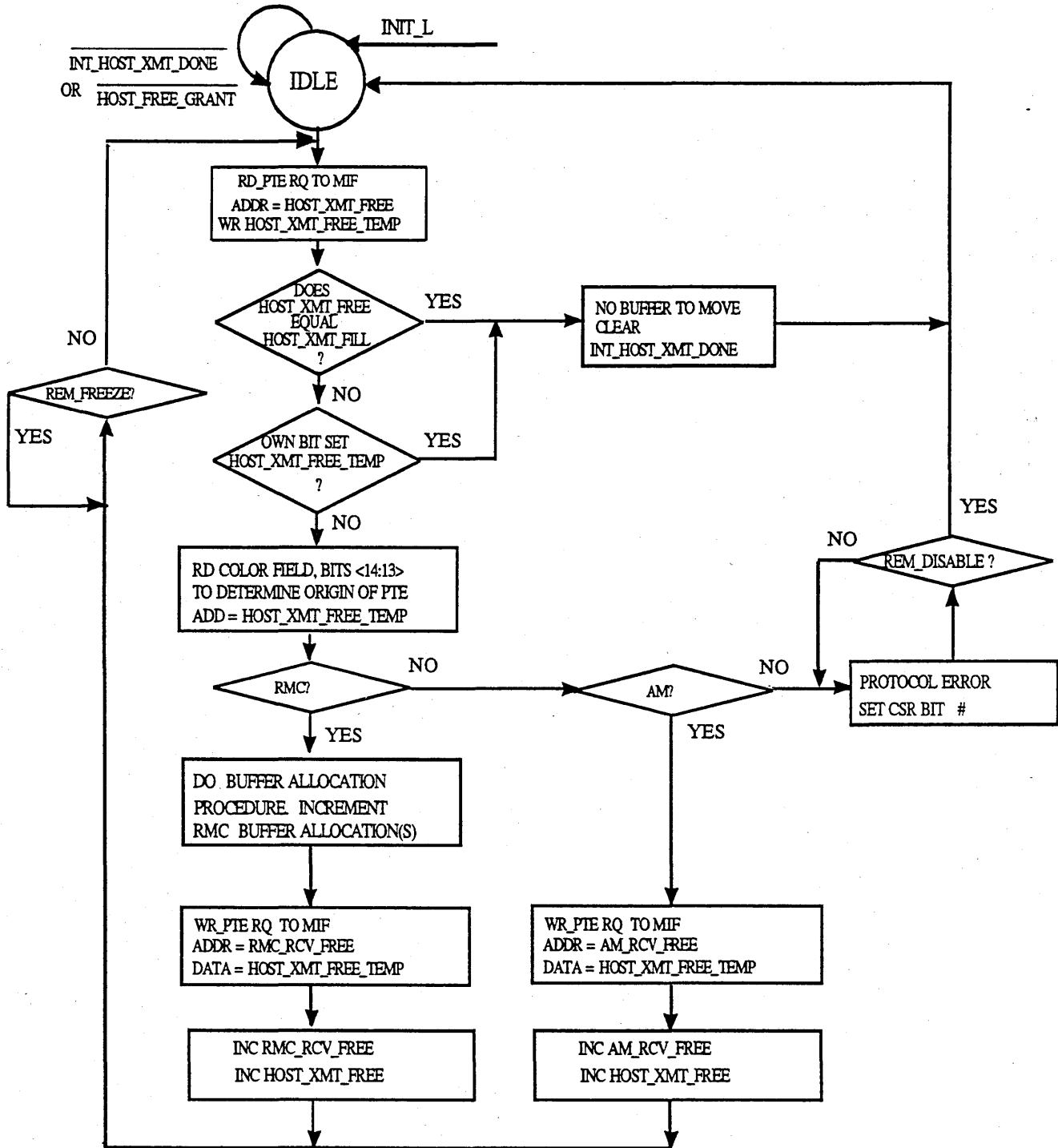
RMC\_MOVE\_FREE FLOW



RMC\_MOVE\_FREEUIS

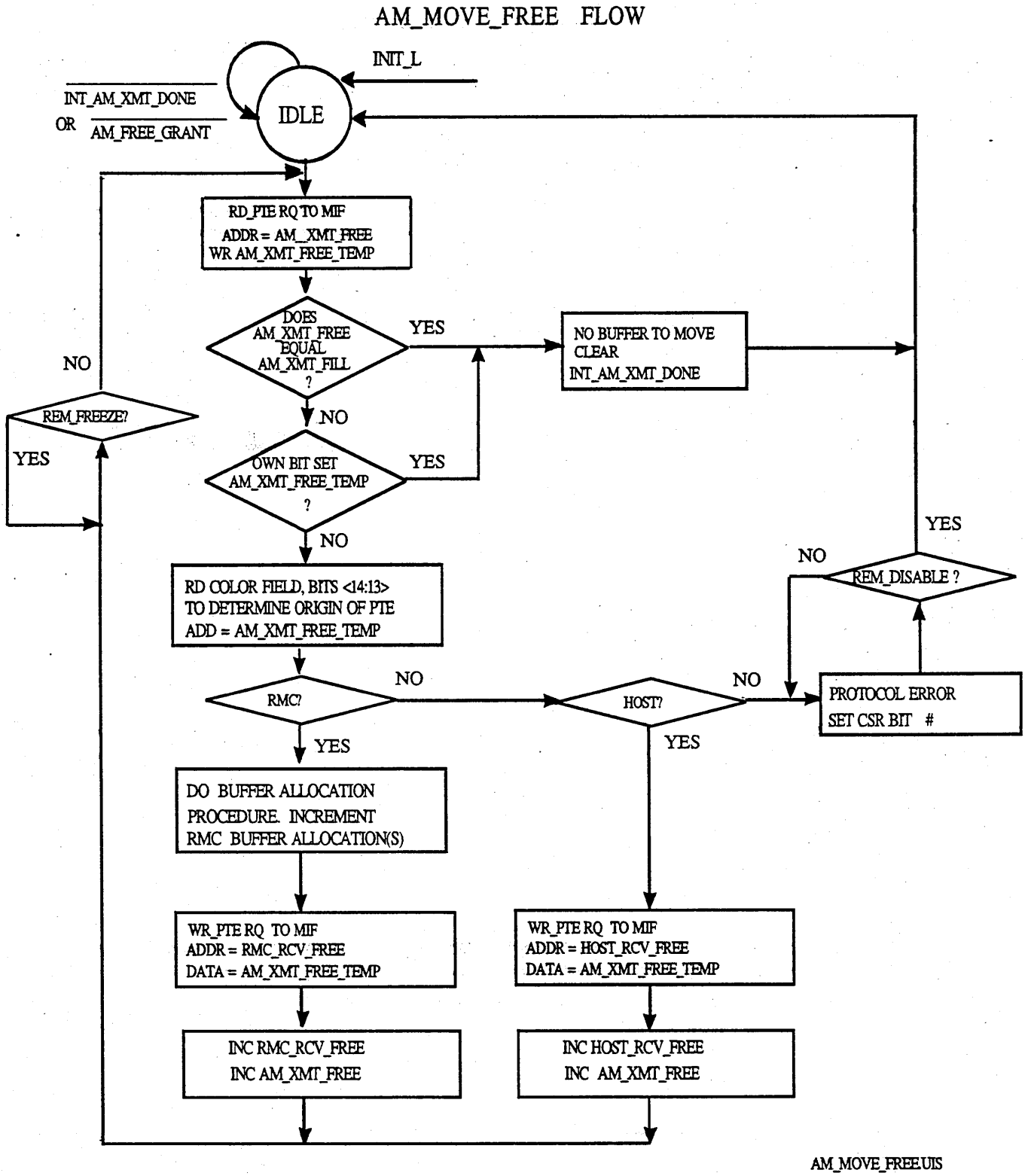
Figure 5-12: HOST MOVE\_FREE FLOW

HOST\_MOVE\_FREE FLOW



HOST\_MOVE\_FREEUIS

Figure 5-13: AM\_MOVE\_FREE FLOW



### 5.8.3 ARBITER LOGIC BLOCK

The REM does not interface directly with the Page Table. Any requests to do reads from or writes to the Page Table will be forwarded to the Memory Interface (MIF) block of the Packet Memory Controller (PMC) gate array.

The MOVE\_FILL and MOVE\_FREE state machines will be expected to be operating in parallel. Both state machines may need to access the Page Table at the same time. Because there is only one bus, there will be an Arbiter Logic block in the REM with an Arbiter State Machine. The Arbiter logic will determine who shall gain access to the bus and select the corresponding request-type, address, and/or data.

Figure 5-14: DIAGRAM OF THE ARBITER STATE MACHINE

### ARBITER INPUT/OUTPUT SIGNALS

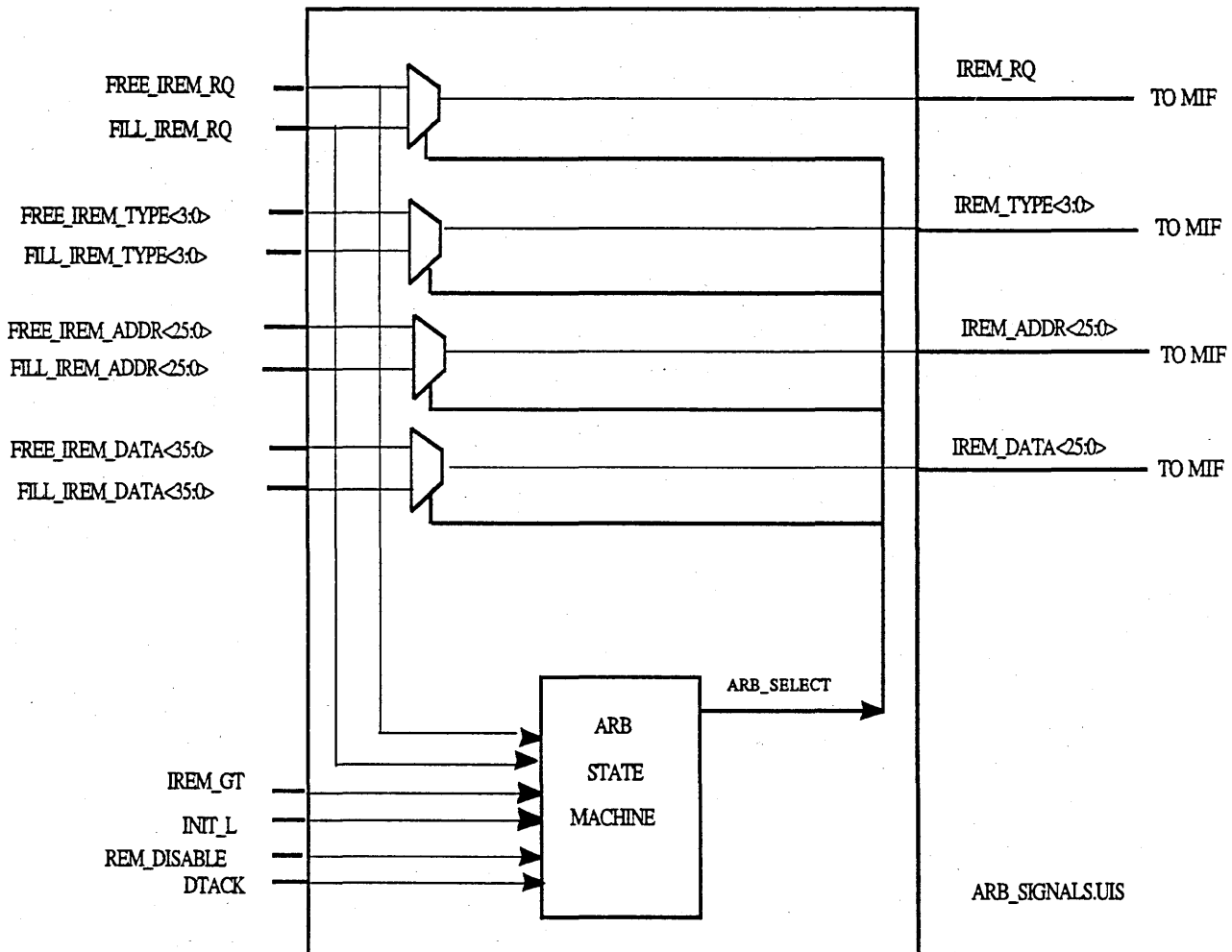
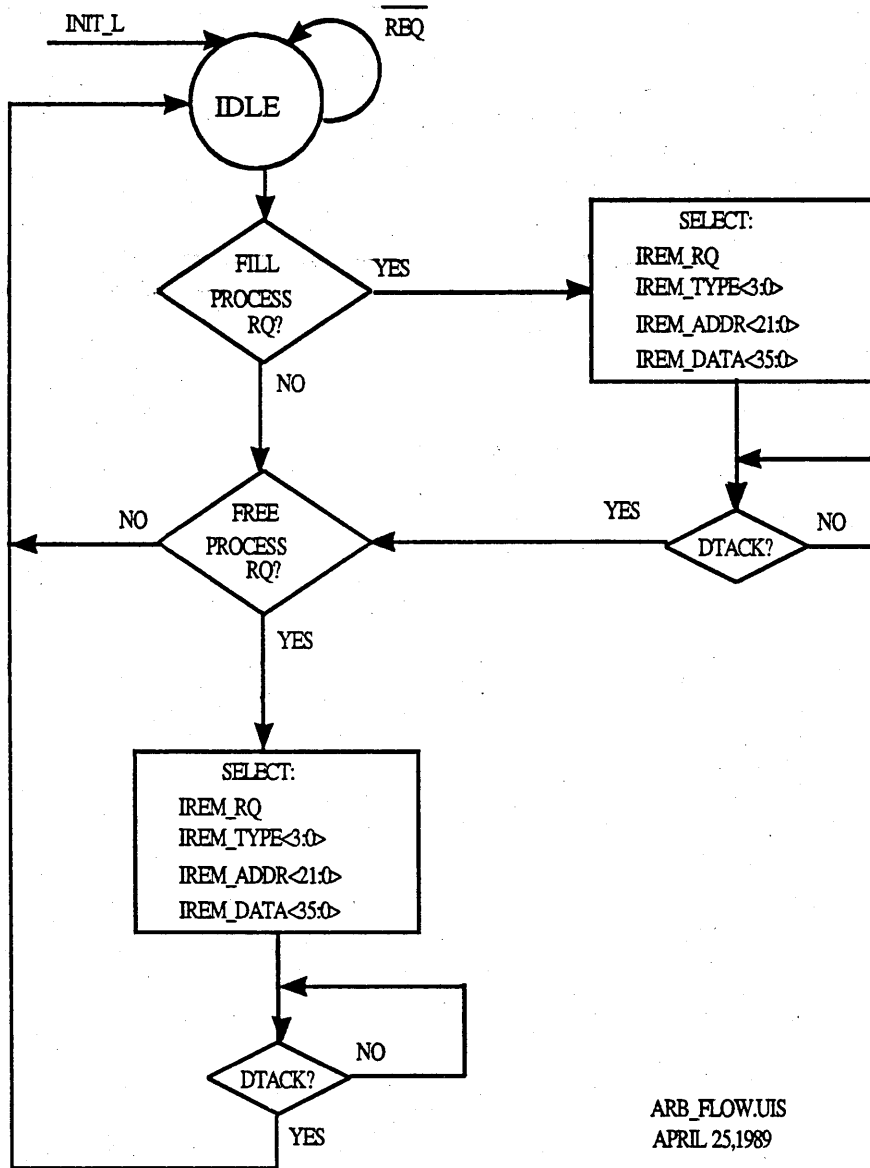


Figure 5-15: ARBITER FLOW

### ARBITER FLOW



ARB\_FLOW.UIS  
APRIL 25,1989

## 5.9 PACKET TRACKING

Upon receipt of a Packet from the FDDI the RMC will then write the Packet into the Packet Buffer Memory with the help of the MIF. Once the Packet has been written to the PBM the RMC will send RCV\_DONE directly to the REM. The REM may have received a RCV\_DONE from the ESP or the AM prior to the RCV\_DONE from the RMC. Therefore, the REM will not respond to the RCV\_DONE from the RMC immediately. In fact, the REM may receive more than one RCV\_DONE from the RMC or from any other interface before it has a chance to respond. The REM must keep track of every RCV\_DONE received from each interface. This is important because the REM MOVE\_FILL state machine moves the PTEs of one Packet only in response to RCV\_DONE. Therefore, to ensure that all Packets that are written to the Packet Buffer Memory are transmitted, the REM must be aware of all RCV\_DONE signals coming in from all three interfaces.

The mechanism that will be implemented to ensure that the REM is aware of all Packets written to the PBM is described here.

- There will be three JK flip flops, one for each interface, whose Q output will be internal\_source\_receive\_done (INT\_RMC\_RCV\_DONE, INT\_ESP\_RCV\_DONE, OR INT\_AM\_RCV\_DONE). When one of the interfaces sends RCV\_DONE it will set the JK flipflop.
- Internal\_source\_receive\_done will clear only when the JK flop is reset.
- INT\_SRC\_RCV\_DONE will transition the MOVE\_FILL Supervisory State Machine from the IDLE state to the Next State.
- If INT\_SRC\_RCV\_DONE is true and a signal called FILL\_GRANT is true, the MOVE\_FILL state machine will be activated.
- The MOVE\_FILL logic will check to see if there are any buffers remaining on the RCV ring. If there are not, obviously, there are also no Packets to be serviced. The check is accomplished by comparing the addresses in the SRC\_RCV\_FREE and SRC\_RCV\_FILL ring registers. If these addresses are equal it indicates that the ring does not have any buffers to be moved. When there are no buffers, PKT\_MOVE\_COMPLETE will be set and CLR\_INT\_SRC\_RCV\_DONE will be asserted.
- CLR\_INT\_SRC\_RCV\_DONE will be directed to an AND gate whose inputs will be CLR\_INT\_RCV\_DONE and NOT SRC\_RCV\_DONE. The output of this AND gate will be the K input of the JK flip flop. When the output is high it will cause a reset, thereby clearing INT\_SRC\_RCV\_DONE. This implementation will take care of the case of both CLR\_RCV\_DONE and RCV\_DONE being true at the same time. In this case, if the Q output was high it would toggle and a Packet would go unprocessed. Therefore, RESET will only occur when RCV\_DONE is not true.
- When INT\_SRC\_RCV\_DONE is not asserted, the MOVE\_FILL state machine will remain in the IDLE state.
- If the addresses are not equal, the MOVE\_FILL logic will check the OWN bit in the PTE that it has written to a REM-local register. If the OWN bit is set, then the REM does not own the PTE and that indicates that there are no more Packets to be moved. As a result, the REM will proceed as in the previous situation. It will CLR\_INT\_SRC\_RCV\_DONE and set PKT\_MOVE\_COMPLETE.

- If the Own bit is clear, then there is a Packet to move and the MOVE\_FILL state machine will continue to follow the process required to move PTEs.

Returning to the description of the REM being off moving Packets for the AM or ESP interfaces while several Packets are being written into the PBM by the RMC and recall that for each Packet the RMC has sent RMC\_RCV\_DONE. Therefore, there are several Packets the REM needs to move from the RMC\_RCV ring once it's the RMC's turn. The initial time that the RMC sent RMC\_RCV\_DONE the JK flip flop was set. Each successive RMC\_RCV\_DONE actually had no affect on the Q output unless the flop was cleared between successive RMC\_RCV\_DONES. The procedure to acknowledge every RCV\_DONE does not require that each one is counted. The REM will move all Packets to be moved as long as INT\_SRC\_RCV\_DONE and FILL\_GRANT are both true. A more detailed explanation follows:

- The REM will follow the procedure listed above to check to see if there are buffers to move and to check for REM ownership. Since the OWN bit is clear the MOVE\_FILL process will procede. When the PTEs corresponding to one Packet have been moved, the REM will set PKT\_MOVE\_COMPLETE and return to the supervisor state machine to determine what to do next.
- In the event there are multiple RMC packets to move, the REM will prioritize these over other traffic unless FTP\_REM is asserted. Up to 256 successive RMC may be moved prior to the REM servicing other received packets.
- The supervisory state machine shows that the REM will now check to see if INT\_HOST\_RCV\_DONE or INT\_AM\_RCV\_DONE is true. If the conditions are true to start the HOST\_MOVE\_FILL or AM\_MOVE\_FILL processes then they will be executed before moving the next Packet to be moved in the RMC\_RCV ring.
- When it is the RMC's turn once again, INT\_RMC\_RCV\_DONE will be true because it was not cleared when the last Packet was moved. Now if RMC\_FILL\_GRANT is also true, then the MOVE\_FILL state machine will transition from the idle state. (Please refer to the Supervisory state machine flow to see under what conditions RMC\_FILL\_GRANT will be true).
- The MOVE\_FILL logic will now check once again to see if there are remaining buffers and for REM ownership. In this case we know that there will be buffers available and the REM will own the next PTE to be moved.
- The REM will move the PTEs and set PKT\_MOVE\_COMPLETE and return to the IDLE state in the MOVE\_FILL state machine and the supervisory state machine will check to see if there are Packets to be moved from the HOST\_RCV ring or the AM\_RCV ring.
- When it is the RMC's turn again, the check to see if there are buffers that need to be moved, may indicate that there aren't. Also, if there are buffers, the REM may not have ownership. If either are true, the REM will set CLR\_INT\_RMC\_RCV\_DONE which will reset the JK flop if SRC\_RCV\_DONE has not come true again. A RESET will deassert INT\_RMC\_RCV\_DONE. The MOVE\_FILL state machine will then transition to the IDLE state.

Upon receipt of a XMT\_DONE from one of the three interfaces the REM will need to return the free buffers from the Packet that was just read out of the Packet Buffer Memory to the RCV ring of origin. The REM is required to return all free buffers and therefore will need to employ a mechanism which indicates when there are more free buffers to return from the XMT ring to the RCV ring. The mechanism is very similar to that outlined above for



keeping track of the Packets that must be moved from the RCV rings. The differences are listed as follows:

- When INT\_RMC\_XMT\_DONE and RMC\_FREE\_GRANT are both true, the ring is not empty and the REM has ownership, the RMC\_MOVE\_FREE state machine will move up to 8 buffers that it owns, regardless of how many Packets are represented, to the RCV ring(s) of origin.
- If there were less than 8 buffers to move, DONE will then be set.
- CLR\_INT\_RMC\_XMT\_DONE will be set and will be an input to an AND gate. The other input to the AND gate will be NOT SRC\_XMT\_DONE. The output of the AND gate will be the K input to the Flop. Hence, the Flop will only RESET if CLR\_XMT\_DONE is true AND XMT\_DONE is not true ensuring that all PTE(s) to be moved, will be. For this example we'll assume that the Flop was RESET.
- When the MOVE\_FREE Supervisory state machine checks for RMC\_INT\_XMT\_DONE when it's the RMC'S turn again, it will only be true if yet another RMC\_XMT\_DONE has been asserted since the MOVE\_FREE process last executed or there were originally 8 or more free buffers to move.
- The check for free buffers and ownership will be repeated as the last step before exiting the MOVE\_FREE state machine. If there are no PTE(s) to move, CLR\_INT\_XMT\_DONE will be asserted.

## 5.10 REM FREEZE AND REM DISABLE

REM\_ENABLE must be asserted for the state machines to run. It is a CSR bit which is set by the Adapter Manager. When REM\_Enable is CLEAR it will be referred to as REM\_DISABLE. REM\_DISABLE will completely shut down the REM. REM\_DISABLE will be set by the Adapter Manager or as a result of a fatal error. All protocol errors will be considered fatal errors. A protocol error will result in the writing of the appropriate error bit in the CSR register. The CSR interface will set a signal called PRO\_ERR which in the following cycle will force all the REM state machines to the IDLE state.

REM\_FREEZE is a signal sent to the REM from the MIF which 'gracefully' freezes the MOVE\_FILL and MOVE\_FREE state machines in the REM. A REM\_FREEZE may be asserted as a result of a request from Station Management(SMT) which may need to stall activity so that it can concentrate on its own activities. REM\_FREEZE is not considered a fatal or irrecoverable condition. The impact of a REM\_FREEZE on the MOVE\_FILL and MOVE\_FREE state machines is described below. When REM\_FREEZE is deasserted the state machines will pick up where they left off.

- MOVE\_FILL: Finish moving the current Packet and stop.
- MOVE\_FREE: Finish moving the current PTE and stop.

## 5.11 TWO EXCEPTION CASES

There are two exception cases to the way that PTEs are generally moved between rings in the Page Table. The Adapter Manager is responsible for implementing the procedure of the exception cases. In both cases a Packet is received by the RMC and is written into the PBM. The HOST bit of the forwarding vector for the PTE associated with this Packet indicates that this Packet is an AM Packet. The REM sends DESTINATION\_START\_XMT to the AM as it would normally. The AM follows the procedure to read the Packet from the PBM. As the AM begins reading the Packet header, it sees that the Packet is not an AM Packet. Instead, the Packet is either a MOP Packet whose destination is the HOST or a SMT Packet whose destination is the RMC. The AM follows the procedure summarized below in this situation:

- Determine the number of free buffers remaining in the FILL section of the AM\_RCV ring.
- If the number of pages (free buffers) needed exceeds the number of buffers available, the AM will check to see if it has free buffers from its own private reserve. If it does not have an adequate number of free buffers, it will copy the Packet to the Adapter Manager's local memory and send AM\_XMT\_DONE to the REM.
- If there are at least the number of free buffers required, proceed.
- Read the base address of the AM\_RCV ring from a register local to the AM.
- Read the base address of the AM\_XMT ring from an address local to the AM.
- Copy the PPN's for the exception Packet from the AM\_XMT ring to the AM\_RCV ring.
- Copy an equivalent number of PPNs from the AM\_RCV ring to the AM\_XMT ring. Copy the PTEs to the same address(es) in the AM\_XMT ring that the PTEs corresponding to the Packet were located in.
- In simple terms swap the the PPNs in the AM\_XMT ring corresponding to the exception Packet, for the corresponding PPNs in the AM\_RCV ring.
- AM clears the own bits for all the PTEs involved in this swap ensuring that the OWN bit of the first PTE is cleared last.
- AM sends AM\_RCV\_DONE and AM\_XMT\_DONE to the REM

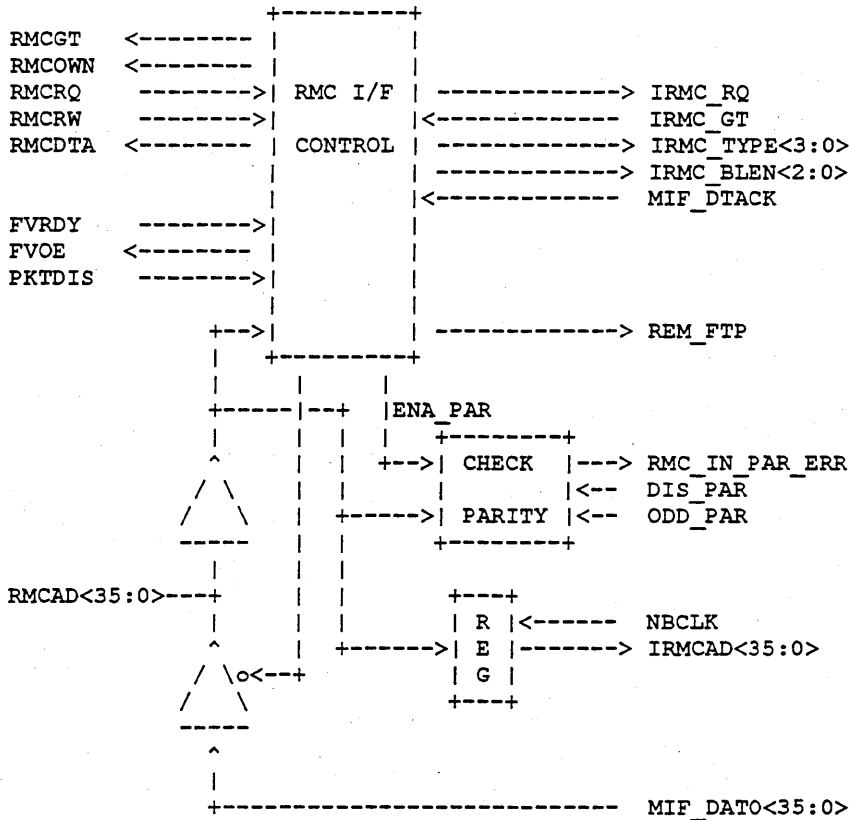
The REM carries out the procedure outlined for the FILL process in response to AM\_RCV\_DONE and carries out the FREE process in response to AM\_XMT\_DONE. The fact that this is an exception Packet is transparent to the REM. The REM responds in the usual fashion to the SOURCE\_RCV\_DONE and DESTINATION\_XMT\_DONE commands. The Adapter Manager has to handle the manipulation of the PTEs for the exception case because the own bits are set for the PTEs in the AM\_RCV ring and the PTEs of the AM\_XMT ring corresponding to the exception Packet. Also, it should be evident that only PTEs have been shuffled during the handling of the exception case and not buffers in the Packet Buffer Memory.

## 5.12 ADDITIONAL REM FEATURES

- The REM can be 'gracefully' frozen. Free buffers or Packets in transit will be allowed to complete and status will be provided on completion.
- Packets written to the PBM by the Host that are RMC-bound by default, can be forced to the Adapter Manager via a CSR bit.
- Packets written to the PBM by the RMC that would normally be headed for the Host or the Adapter Manager can be forced only to the host or only to the Adapter Manager via CSR bits.
- The following are provided to the XFA/Host Interface:
  - Status indicating that there are eighteen or more free buffers in the HOST\_RCV ring.



Figure 6-2: RMC INTERFACE BLOCK DIAGRAM



## 6.2 SPECIAL FEATURES OF THE RMC INTERFACE

The RMC INTERFACE has the ability to cache RMC\_RCV Write Buffer Descriptor and RMC\_RCV Clear Own instructions and apply them to the memory subsystem at a later time. These features are enabled when the IFVRDY (Ignore Forwarding Vector Ready) bit in CSR00 is deasserted. This is to allow the Parser extra time to complete its calculation of the Forwarding Vector. Any RMC\_RCV WBD instruction is automatically cached in the interface. Any successive RMC\_RCV WBD instruction will overwrite the previous WBD. When the interface sees an RMC\_RCV CO instruction it caches the instruction and waits for FVRDY from the PARSER gate array to assert. Upon assertion, the RMCIF will deliver the cached WBD and CO instructions to the MIF back to back. It should be noted that the RMC\_RCV WBD and CO instructions are responded to immediately by the RMCIF so the RMC believes that these instructions have been honored. so it may issue another instruction. This instruction will not be granted until the cached WBD and CO instructions have been internally processed.

It is system architecture that RMC\_XMT buffer descriptors are not returned to the host so the information that the RMC returns in them after transmitting a packet is not normally used. It is useful to have this information for debug purposes. The RMCIF has the ability to selectively not deliver RMC\_XMT buffer descriptors. If IRXWBD bit (Ignore Rmc Write Buffer Descriptor) is asserted in CSR00, the RMCIF responds to the RMC as if the instruction had been processed but does not deliver the instruction to the MIF. This frees the MIF to do more useful transactions.

If the PARSER gate array asserts a signal called PKTDIS this indicates to the RMCIF that the PARSER will be discarding the packet currently being delivered. The RMCIF responds to this signal by not delivering any subsequent RMC\_RCV write data bursts to the MIF. The RMC is responded to as if the instruction had been processed. This frees the MIF to do more useful transactions.

The RMC interface provides the REM with information that it should give priority to packet moves targeted to the RMC Transmit Ring. This occurs while the RMC is actively transmitting. The algorithm is as follows:

### Figure 6-3: REM Force Transmit Priority Algorithm

#### REM FORCE TRANSMIT PRIORITY ALGORITHM

-----  
This algorithm SETS or CLEARS FTP\_REM which is READ by the REM'S MOVE\_FILL process state machine. SETTING FTP\_REM will result in the HOST or AM\_RCV\_DONES to be serviced before the RMC\_RCV\_DONES. In this way the filling of the RMC\_XMT ring is prioritized in order to take maximum advantage of any captured token.

```
WHILE INITL THEN FTP_REM = 0;  
WHEN RMC_XMT_RQ THEN FTP_REM = 1;  
WHEN RMC_RCV_RQ THEN FTP_REM = 1;
```

The RMCIF contains logic to support the BOOKMARK function. A logic function monitors all RMC\_RCV traffic in order to determine at what address the RMC is delivering receive traffic. This address is saved. On CSR enabled request (BMRKRQ), the last saved address is registered and compared to the corresponding REM move fill pointer. When the saved address and the REM pointer match, the RMCIF sets status causing a bookmark interrupt. This function ensures that all entries in the RMC\_RCV queue at the time of the bookmark request have been delivered prior to the interrupt.

The RMCIF contains logic to ensure that lockout cannot occur in the event that the PARSER gate array fails to deliver a forwarding vector. At any time the RMCIF expects to get a FVRDY indication from the PARSER a counter is started from 0. If FVRDY is seen, the counter is reset. If the counter increments to 15, status is passed resulting in a FVTIMEOUT interrupt.

It should be noted that for test purposes the RMCIF can process PTE read and write requests. Refer to the chapter on the ESP interface for details.







Figure 7-3: SYNCHRONIZER BLOCK DIAGRAM

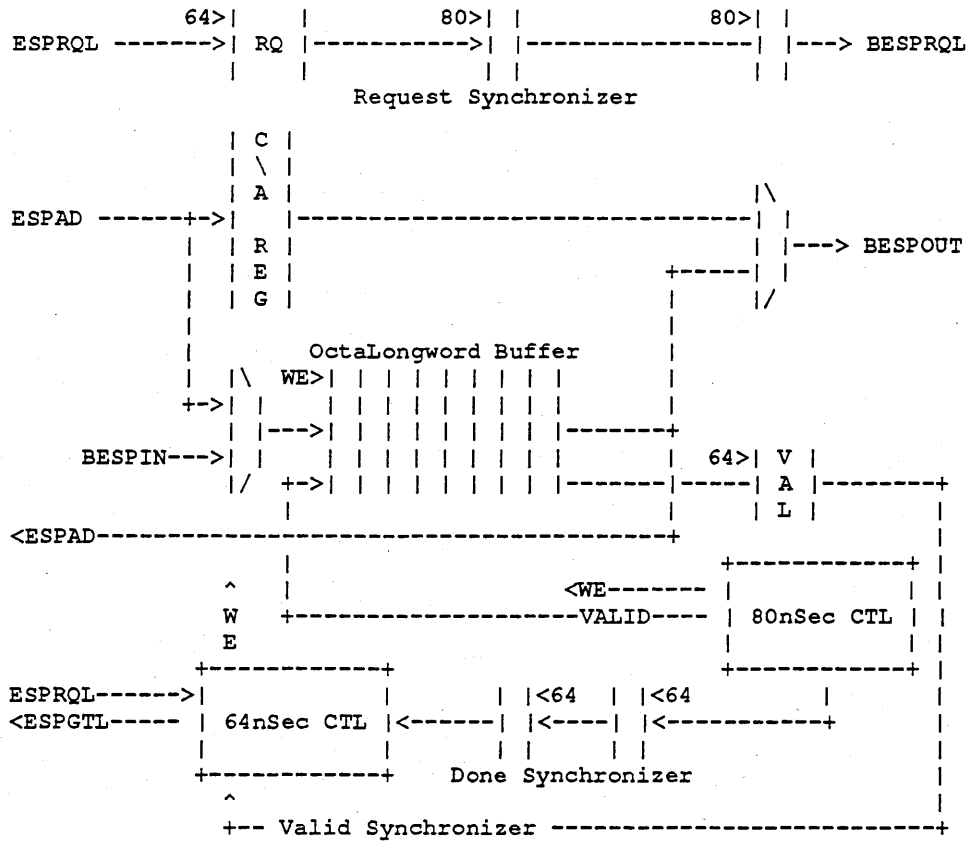
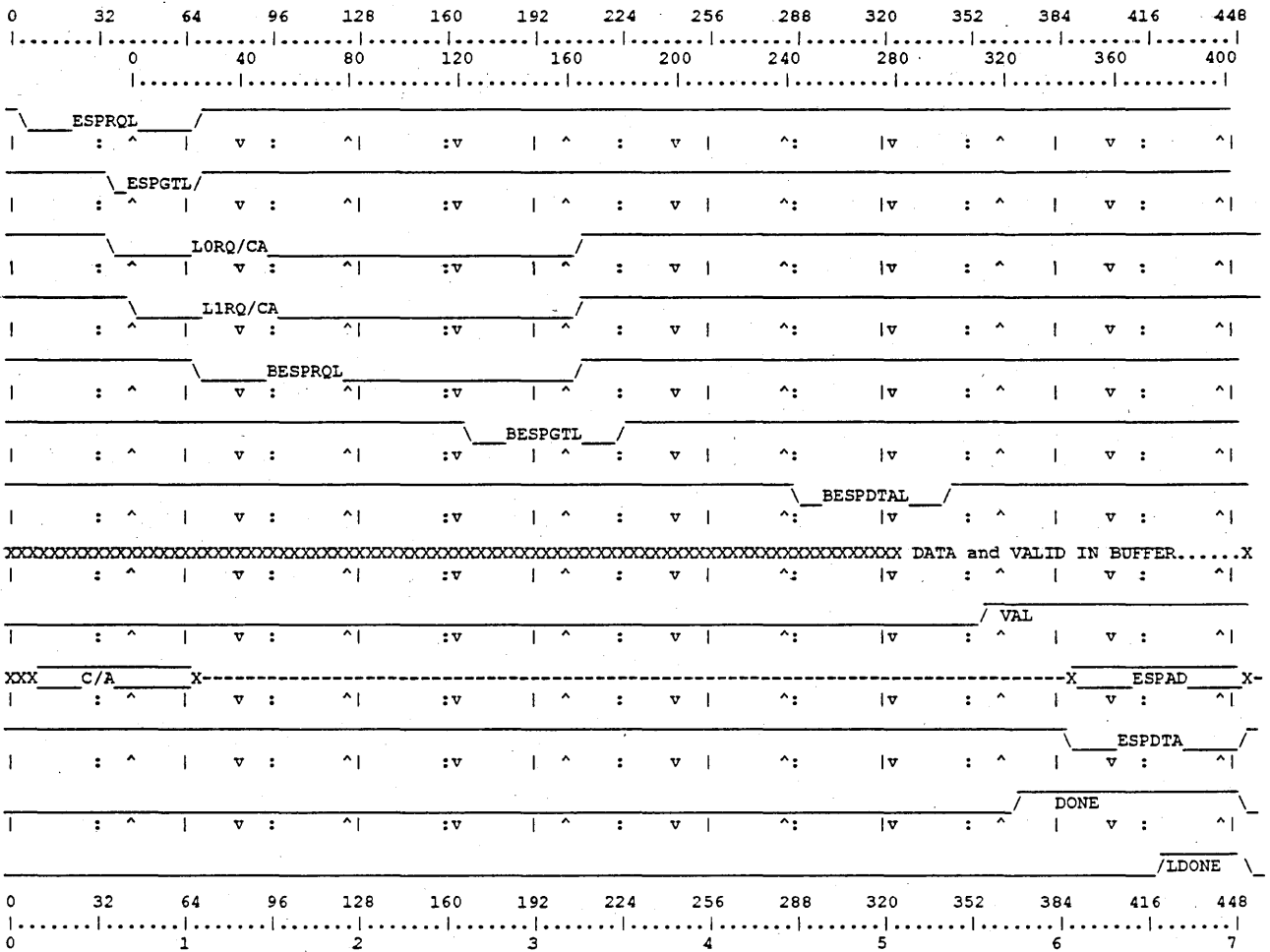


Figure 7-4: BASIC SYNCHRONIZER TIMING

Example shows Buffer Descriptor Read, no xlate:



### 7.1.2 THE ESP INTERFACE FUNCTIONAL BLOCK

The ESPIF block is essentially identical to the RMCIF (refer to the chapter on the RMCIF) with the following caveats:

- **HOST\_XMT\_BD** read with a length field of ^B001 will give two data cycles, the first being the associated page table entry and the second being the BD.
- **HOST\_RCV\_BD** write with implied clear own will be honored (implied clear owns on the RMCIF are ignored) and the WBD automatically followed by a clear own without an external command.
- **ESPAD<24>** The meaning of this bit in a Command/Address Cycle is different from the RMC definition. Here, if set, the command is *NOT* a regular RMC instruction and requires special interpretation.
- **ESPAD<27>** The meaning of this bit in a Command/Address Cycle is different from the RMC definition. Here, if set, the command is associated with the Adapter Manager else it is a Host Ring Command.

### 7.1.2.1 SPECIAL CYCLES

The following cycles are in addition to the regular RMC instructions.

- Read / write CSR
- Read / write PTE

It should be noted that PTE writes AUTOMATICALLY SET THE OWN BIT

Figure 7-5: CSR CYCLE FORMAT

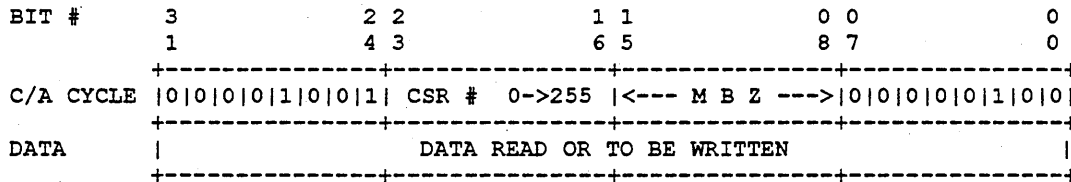
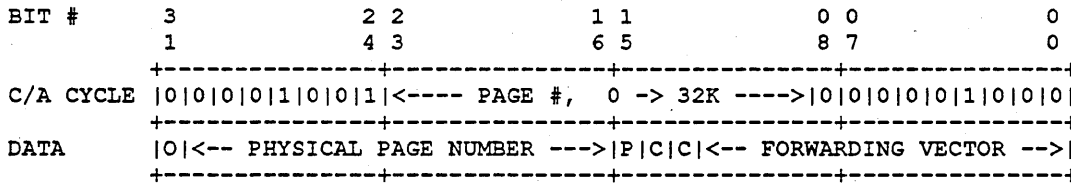


Figure 7-6: PTE CYCLE FORMAT



WHERE: O IS THE OWNERSHIP BIT ASSOCIATED WITH THE PAGE TABLE ENTRY  
This bit is read only, writing has no effect.

WHERE: P IS THE FORWARDING VECTOR PARITY BIT  
This bit is read only, writing has no effect.

WHERE: CC REPRESENTS THE 'COLOR' OF THE BUFFER

- COLOR - 00 = INVALID
- 01 = RMC Receive Ring
- 10 = AM Receive Ring
- 11 = Host Receive Ring

### 7.1.3 THE CSR FUNCTIONAL BLOCK

The CSR functional block is functionally part of the synchronizer in that it provides the 80nSec time domain control to the SYNC block for CSR requests from the ESP gate array. This block contains all of the global control to the PMC through CSR00 and all of the interrupt control and status through CSR02. 32 CSR's are accessible and any address out of that range will result in a Non-Existant CSR interrupt unless masked. The addresses of the available CSR's are 0 through 31. This block controls the read selection and the write enables for all the PMC CSR's. Please note that detained descriptions of the CSRs are contained in a separate chapter.

Figure 7-7: CSR I/F FUNCTIONAL BLOCK DIAGRAM

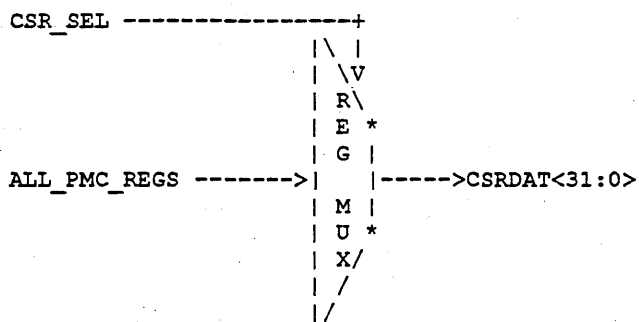
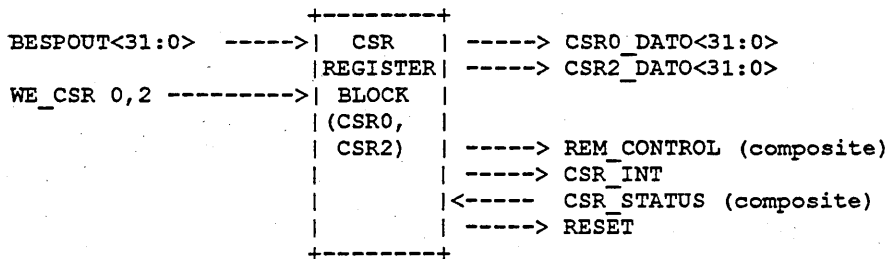
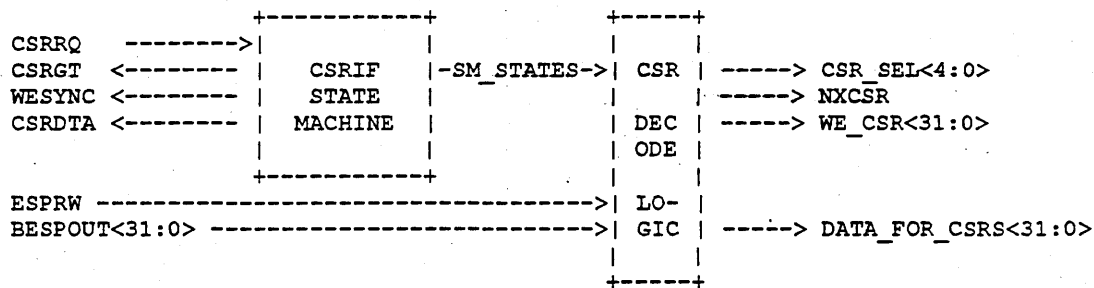
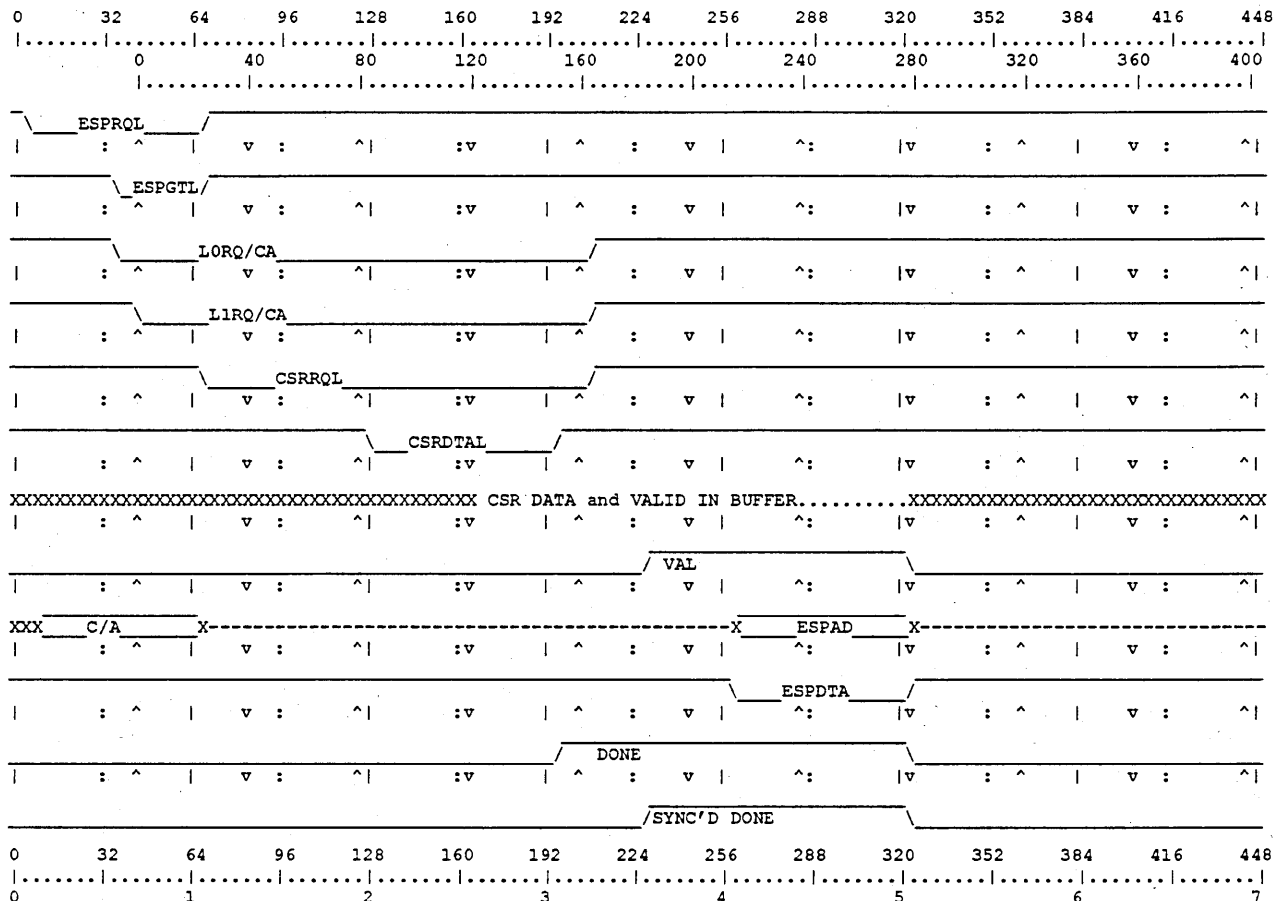


Figure 7-8: CSR COMMAND BLOCK TIMING

Example shows CSR Read:



# CHAPTER 8

## PMC REGISTERS

This chapter describes all the registers available within the PMC gate array. They are visible only by the 68020.

Conventions:

All address values stated are the offset from the PMC CSR base address.

All undefined bits are NIO (Not Implemented, read as 0)

All register bits are write 1 to set, write 0 to clear unless otherwise stated.

Accessing a non-existent register in PMC CSR register block will return the value of the register accessed by the low 5 binary bits of the CSR accessed and cause an interrupt, unless masked.

### 8.1 PMC REGISTER SUMMARY

Following is a summary of the PMC CSR REGISTER BLOCK:

**Table 8-1: PMC CSR REGISTER ALLOCATION**

| ADDR             | BITS   | NAME                                      | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|--------|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>CSR BLOCK</i> |        |                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ^X00             | <31:0> | PMC Control and Status Register           | This is the register of primary control for the PMC. See full definition elsewhere in this chapter.                                                                                                                                                                                                                                                                                                                                                                 |
| ^X01             | <31:0> | PMC RAM Parity Error Register             | In the event of a MIF parity error (see definition in CSR00), this register contains the address of the first detected address with data in error plus information identifying which parity protected data field(s) was in error. Under non-error circumstances this register is updated with the address of every read operation. This register can be written (for test purposes) only when REM_ENABLE is NOT set. See full definition elsewhere in this chapter. |
| ^X02             | <31:0> | PMC Interrupt and Interrupt Mask Register | In the event of a PMC interrupt, this register provides the reason(s) for the interrupt. All interrupts are maskable; the mask bits are in this register. See full definition elsewhere in this chapter.                                                                                                                                                                                                                                                            |
| ^X03             | <31:0> | PMC REM Protocol Error Register           | In the event that bit 4 of CSR02 is set (REM Protocol Error Summary Bit), this register provides full details of which protocol error(s) the REM encountered. See full definition elsewhere in this chapter.                                                                                                                                                                                                                                                        |

**Table 8–1 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR                        | BITS    | NAME                         | DEFINITION                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|---------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>REM RMC QUEUES BLOCK</i> |         |                              |                                                                                                                                                                                                                                                                                                                                                             |
| ^X04                        | <14:0>  | RMC_RCV_FIL<br>POINTER       | AM programmed as the base of the RMC_RCV ring, read as current RMC_RCV_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                          |
| ^X04                        | <30:16> | RMC_RCV_FREE<br>POINTER      | AM programmed as the first location in the RMC_RCV ring at which free space should be put by the REM (eg: RMC_RCV ring is 1024 entries long and is initialized with 500 buffers. RMC_RCV_FREE pointer is initialized at the 501st location). Read as current RMC_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                           |
| ^X04                        | <31:0>  | PMT Data Register 0          | This register doubles as the Packet Memory Test Data Register 0. It initializes as ^XAAAAAAAA and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                                                   |
| ^X05                        | <14:0>  | RMC_XMT_FIL<br>POINTER       | AM programmed as the base of the RMC_XMT ring, read as current RMC_XMT_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                          |
| ^X05                        | <30:16> | RMC_XMT_FREE<br>POINTER      | AM programmed as the base of the RMC_XMT ring, read as current RMC_XMT_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                         |
| ^X05                        | <31:0>  | PMT Data Register 4          | This register doubles as the Packet Memory Test Data Register 4. It initializes as ^X33333333 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                                                   |
| ^X06                        | <14:0>  | RMC_RCV_SIZE                 | AM programmed with the last address of the RMC_RCV ring and can only be written when REM_ENABLE is NOT set. The number of entries in the RMC_RCV ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                                              |
| ^X06                        | <30:16> | RMC_XMT_SIZE                 | AM programmed with the last address of the RMC_XMT RING and can only be written when REM_ENABLE is NOT set. The number of entries in the RMC_XMT ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                                              |
| ^X07                        | <31:0>  | PMT Failing Address Register | In the event that Packet Memory Test fails, this register will contain the failing address and some other information. While PMT is active, this register contains the current access address. See full definition elsewhere in this chapter. Initialized as 0 but may not be seen at this value since power-up PMT or programmed PMT may change this value |
| <i>REM AM QUEUES BLOCK</i>  |         |                              |                                                                                                                                                                                                                                                                                                                                                             |
| ^X08                        | <14:0>  | AM_RCV_FIL<br>POINTER        | AM programmed as the base of the AM_RCV ring, read as current AM_RCV_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                                            |
| ^X08                        | <30:16> | AM_RCV_FREE<br>POINTER       | AM programmed as the first location in the AM_RCV ring at which free space should be put by the REM. Read as current AM_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                    |

**Table 8–1 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR | BITS    | NAME                      | DEFINITION                                                                                                                                                                                                                                                                                                                          |
|------|---------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X08 | <31:0>  | PMT Data Register 1       | This register doubles as the Packet Memory Test Data Register 1. It initializes as ^X55555555 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                           |
| ^X09 | <14:0>  | AM_XMT_FIL POINTER        | AM programmed as the base of the AM_XMT ring, read as current AM_XMT_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                    |
| ^X09 | <30:16> | AM_XMT_FREE POINTER       | AM programmed as the base of the AM_XMT ring, read as current AM_XMT_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                                                                                                                                   |
| ^X09 | <31:0>  | PMT Data Register 5       | This register doubles as the Packet Memory Test Data Register 5. It initializes as ^X00000000 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                           |
| ^X0A | <14:0>  | AM_RCV_SIZE               | AM programmed with the last address of the AM_RCV ring and can only be written when REM_ENABLE is NOT set. The number of entries in the AM_RCV ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                        |
| ^X0A | <30:16> | AM_XMT_SIZE               | AM programmed with the last address of the AM_XMT ring and can only be written when REM_ENABLE is NOT set. The number of entries in the AM_XMT ring must be a power of 2. Can also be read. Initialized as 0                                                                                                                        |
| ^X0B | <31:0>  | PMT Failing Data Register | In the event that Packet Memory Test fails, this register will contain the failing data pattern. While PMT is active, this register contains the current data pattern. See full definition elsewhere in this chapter. Initialized as 0 but may not be seen at this value since power-up PMT or programmed PMT may change this value |

*REM HOST QUEUES BLOCK*

|      |         |                       |                                                                                                                                                                                                                           |
|------|---------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X0C | <14:0>  | HOST_RCV_FIL POINTER  | AM programmed as the base of the HOST_RCV ring, read as current HOST_RCV_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                      |
| ^X0C | <30:16> | HOST_RCV_FREE POINTER | AM programmed as the first location in the HOST_RCV ring at which free space should be put by the REM. Read as current HOST_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                              |
| ^X0C | <31:0>  | PMT Data Register 2   | This register doubles as the Packet Memory Test Data Register 2. It initializes as ^X99999999 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail |
| ^X0D | <14:0>  | HOST_XMT_FIL POINTER  | AM programmed as the base of the HOST_XMT ring, read as current HOST_XMT_FIL pointer, can only be written when REM_ENABLE is NOT set                                                                                      |
| ^X0D | <30:16> | HOST_XMT_FREE POINTER | AM programmed as the base of the HOST_XMT ring, read as current HOST_RCV_FREE pointer, can only be written when REM_ENABLE is NOT set                                                                                     |



**Table 8-1 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR | BITS    | NAME                            | DEFINITION                                                                                                                                                                                                                                                                |
|------|---------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X0D | <31:0>  | PMT Data Register 6             | This register doubles as the Packet Memory Test Data Register 6. It initializes as ^X00000000 and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                 |
| ^X0E | <14:0>  | HOST_RCV_SIZE                   | AM programmed with the last address of the HOST_RCV ring and can only be written when REM_ENABLE is NOT set. The number of entries in the HOST_RCV ring must be a power of 2. Can also be read. Initialized as 0                                                          |
| ^X0E | <30:16> | HOST_XMT_SIZE                   | AM programmed with the last address of the HOST_XMT ring and can only be written when REM_ENABLE is NOT set. The number of entries in the HOST_XMT ring must be a power of 2. Can also be read. Initialized as 0                                                          |
| ^X0F | <31:0>  | PMT Control and Status Register | This register controls and provides some status concerning the Packet Memory Test. See full definition elsewhere in this chapter. Initialized as ^X04000011 at power up only but may not be seen at this value since power-up PMT or programmed PMT may change this value |

*ALLOCATION REGISTERS*

|      |         |                              |                                                                                                                                                                                                                                                                                                                          |
|------|---------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X10 | <14:0>  | RMC_HOST ALLOCATION REGISTER | Written with the total number of RMC_FREE buffers to be allocated to the RMC/HOST path and can only be written when REM_ENABLE is NOT set. Read as the current count of available RMC to Host buffers. Initialized as 0                                                                                                  |
| ^X10 | <30:16> | RMC_AM ALLOCATION REGISTER   | Written with the total number of RMC_FREE buffers to be allocated to the RMC/AM path and can only be written when REM_ENABLE is NOT set. Read as the current count of available RMC to AM buffers. Initialized as 0                                                                                                      |
| ^X11 | <7:0>   | RMC_SMT OVERDRAFT REGISTER   | Written with the total number of RMC_FREE buffers to be allocated as an overdraft for the RMC/AM path, specifically for SMT packets and only when RMC_AM does not have enough buffers and can only be written when REM_ENABLE is NOT set. Read as the current count of available SMT overdraft buffers. Initialized as 0 |
| ^X11 | <15:8>  | RMC_MOP OVERDRAFT REGISTER   | Written with the total number of RMC_FREE buffers to be allocated as an overdraft for the RMC/AM path, specifically for MOP packets and only when RMC_AM does not have enough buffers and can only be written when REM_ENABLE is NOT set. Read as the current count of available MOP overdraft buffers. Initialized as 0 |
| ^X11 | <30:16> | RMC_ERR ALLOCATION REGISTER  | Written with the maximum number of error buffers that may be in circulation at any given time using RMC_FREE buffers and can only be written when REM_ENABLE is NOT set. Read as the current count of error buffers that may be delivered. Initialized as 0                                                              |

*FREE BUFFER COUNTERS*

**Table 8–1 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR                             | BITS    | NAME                   | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------|---------|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X12                             | <30:16> | HOST_FREE COUNTER      | AM programmed as number of buffers allocated at initialization to the HOST_RCV queue. Read as the current count of free buffers available for the HOST_RCV queue. Can only be written when REM_ENABLE is NOT set. Initialized as 0                                                                                                                                                                                                                    |
| <i>TRAFFIC CONTROL REGISTERS</i> |         |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| ^X13                             | <7:0>   | RMC backed up variable | If the RMC interface sees this number of RMC back-to-back requests, the ESP interface will be denied service. These bits are WRITE 1 TO SET AND WRITE 1 TO CLEAR and initialized as ^XFF                                                                                                                                                                                                                                                              |
| ^X13                             | <15:8>  | ESP lockout variable   | If the RMC interface considers the RMC to be backed up, the ESP will be denied service for this number of RMC back-to-back requests, before the ESP will be allowed service. These bits are WRITE 1 TO SET AND WRITE 1 TO CLEAR and initialized as ^XFF                                                                                                                                                                                               |
| <i>STATISTICS COUNTERS</i>       |         |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| ^X14                             | <23:0>  | PARSER DISCARD COUNTER | This counter contains a count of the packets discarded because the PARSER set the discard bit (ie packets not destined for this station). The count is cleared by being read. This counter will interrupt on setting of bit 23 (maskable). This will happen approximately every 10 secs in the case of an infinite stream of minimum sized packets all not for this station. This counter can be written when REM_ENABLE is not set. Initialized as 0 |
| ^X15                             | <15:0>  | HOST DISCARD COUNTER   | This counter contains a count of the HOST packets discarded because no RMC/HOST buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 15 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized LLC packets all destined for the host. This counter can be written when REM_ENABLE is not set.                                              |
| ^X15                             | <31:16> | AM DISCARD COUNTER     | This counter contains a count of the AM (but not SMT or MOP) packets discarded because no RMC/AM buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 31 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized LLC packets all destined for the adapter manager. This counter can be written when REM_ENABLE is not set.                  |
| ^X15                             | <31:0>  | PMT Data Register 3    | This register doubles as the Packet Memory Test Data Register 3. It initializes as ^XCCCCCCC and can only be written when REM_ENABLE is NOT set. Note that if this register is written while PMT is active, PMT may fail                                                                                                                                                                                                                              |
| ^X16                             | <15:0>  | SMT DISCARD COUNTER    | This counter contains a count of the SMT packets discarded because no SMT overdraft buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 15 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized SMT packets all for this station. This counter can be written when REM_ENABLE is not set. Initialized as 0.                             |

**Table 8-1 (Cont.): PMC CSR REGISTER ALLOCATION**

| ADDR | BITS    | NAME                            | DEFINITION                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------|---------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ^X16 | <31:16> | MOP DISCARD COUNTER             | This counter contains a count of the MOP packets discarded because no MOP overdraft buffers were available. The count is cleared by being read. This counter will interrupt on setting of bit 31 (maskable). This will happen approximately every 100 mSecs in the case of an infinite stream of minimum sized MOP packets all for this station. This counter can be written when REM_ENABLE is not set. Initialized as 0.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ^X17 | <15:00> | OVERSIZE PACKET DISCARD COUNTER | This counter contains the count of HOST DESTINED packets discarded because the byte count in the RMC delivered Buffer Descriptor exceeded the value in the MAXIMUM SIZE REGISTER. This counter is cleared by reading this register. This counter will interrupt on setting bit 15 (maskable). This will happen approximately every 2.7 secs in the case of an infinite stream of minimum sized oversize packets all for this station. This counter can be written when REM_ENABLE is not set.                                                                                                                                                                                                                                                                                                                                                                       |
| ^X17 | <31:16> | ERROR DISCARD COUNTER           | This read-only counter contains a count of the ERROR packets discarded because either there were no AM buffers available or the ERROR ALLOCATION has been exceeded. The count is cleared by being read. This counter will interrupt on setting of bit 31 (maskable). This will happen approximately every 100mSecs in the case of an infinite stream of minimum sized error packets all for this station. This counter can be written when REM_ENABLE is not set.                                                                                                                                                                                                                                                                                                                                                                                                   |
| ^X17 | <31:0>  | PMT Data Register 7             | This register doubles as the Packet Memory Test Data Register 7. It initializes as ^X0003C95A and can only be written when REM_ENABLE is NOT set. All PMT data registers other than this one contain the data for bits <31:0> of the DRAM; this register contains the data for bits <35:32> and each nibble is associated with one of the other data registers. The lowest nibble is associated with data register 0, the second lowest nibble with data register 1 and so on to the second highest nibble. The highest nibble is always read as 0 and not associated with any data. As an example to make the zeroth pattern all 1's one would write PMT data register 0 (CSR04) with ^X11111111 and PMT data register 7 (CSR17) bits <3:0> with ^B0001 leaving the other data unchanged. Note that if this register is written while PMT is active, PMT may fail. |
| ^X18 | <12:0>  | MAXIMUM SIZE REGISTER           | AM programmed as the maximum byte count that a host destined packet may have in order to be delivered. Packets exceeding this byte count will be counted and discarded. PLEASE NOTE THAT PACKETS OF ONE PAGE OR LESS CANNOT BE DISCARDED THIS WAY. This register can only be written when REM_ENABLE is NOT set. Initialized as 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| ^X19 | <31:0>  |                                 | Not implemented, read as ^X19191919.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ^X1A | <31:0>  |                                 | Not implemented, read as ^X1A1A1A1A.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ^X1B | <31:0>  |                                 | Not implemented, read as ^X1B1B1B1B                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1C | <31:0>  |                                 | Not implemented, read as ^X1C1C1C1C                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1D | <31:0>  |                                 | Not implemented, read as ^X1D1D1D1D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1E | <31:0>  |                                 | Not implemented, read as ^X1E1E1E1E                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| ^X1F | <31:0>  |                                 | Not implemented, read as ^X1F1F1F1F                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |



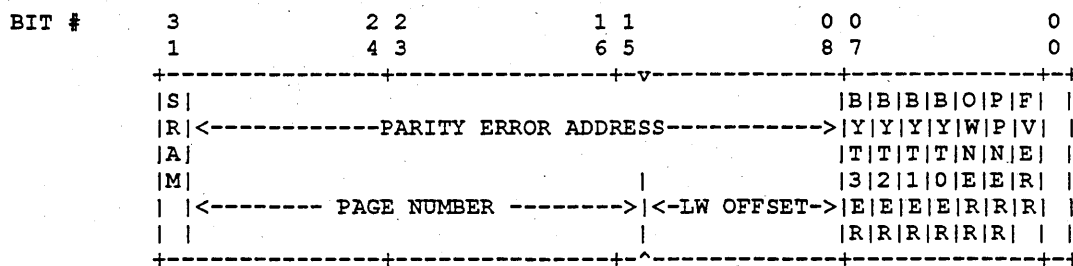
| Bit # | Mnemonic | Definition                                                                                                                                                                                                                                                                                                       |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20    | HSTSTX   | HOST START TRANSMIT: (Only meaningful when SURMOD is asserted.) When written with 1, a pulse is generated which indicates to the ESP that one or more packets have been queued on the HOST TRANSMIT RING. Writing 0 to this bit has no effect. Always read as 0, ie write only                                   |
| 19    | RMCRDN   | RMC RECEIVE DONE: (Only meaningful when SURMOD is asserted.) When asserted, it indicates that one or more packets have been queued on the RMC RECEIVE RING. Writing 1 to this bit, clears. Writing 0 has no effect. Initialized as 0                                                                             |
| 18    | RMCTDN   | RMC TRANSMIT DONE: (Only meaningful when SURMOD is asserted.) When asserted, it indicates that one or more packets have been transmitted from the RMC TRANSMIT RING. Writing 1 to this bit, clears. Writing 0 has no effect. Initialized as 0                                                                    |
| 17    | RMCSTX   | RMC START TRANSMIT: (Only meaningful when SURMOD is asserted.) When written with 1, a pulse is generated which indicates to the RMC that one or more packets have been queued on the HOST TRANSMIT RING. Writing 0 to this bit has no effect. Always read as 0, ie write only                                    |
| 16    | BMRKRQ   | BOOKMARK REQUEST: This is a Write Only bit, read as 0. When written with 1, a pulse is generated which starts the Bookmark State machine. The bookmark state machine ensures that the RMC_RCV queue is flushed up to the packet current at the time of bookmark request and the issues an interrupt (see CSR02). |
| 15    | REMFTP   | REM FORCE TRANSMIT PRIORITY: A read only copy of the internal Force Transmit Priority signal generated by the RMC interface of the PMC.                                                                                                                                                                          |
| 14    | MASKRX   | MASK RMC XMT: When asserted it will mask the the RMCSTX signal normally asserted by the REM.                                                                                                                                                                                                                     |
| 13    | IRXWBD   | When asserted it will cause the RMC Interface to internally discard transmit-write buffer descriptor transactions. The RMC is responded to as if the transaction actually happened.                                                                                                                              |
| 12    | PASPAR   | PASS PARITY: This bit is readable and write 1 to set, write 1 to clear and initialized to 0. When asserted, it will force the passing of internally generated parity, odd or even according to bit 9 of this register, with data from the PBI bus regardless of the parity passed on that bus.                   |
| 11:10 | XERCOD   | This is a read only copy of the AMI fatal error lines. Note that assertion of either XERCOD line causes FINITL to assert. This resets the FDDI chip set and Parser.                                                                                                                                              |
| 9     | EVNPAR   | EVEN PARITY: When asserted, all parity generators and checkers should generate/check even parity                                                                                                                                                                                                                 |
| 8     | DISPAR   | DISABLE PARITY: When asserted, all parity checkers will be disabled, ie parity errors will not be reported.                                                                                                                                                                                                      |
| 7     | IFVRDY   | FORWARDING VECTOR READY: When asserted, RMC_RCV_CLR_OWN transactions will be honored without waiting for the PARSER to assert FVRDY.                                                                                                                                                                             |
| 6     | FRMCXD   | FORCE RMC TRANSMIT DONE: This bit is write only and read as 0. When written with a 1, it will issue an RMC transmit done to the REM as if it had come from the RMC bus.                                                                                                                                          |
| 5     | FRMCRD   | FORCE RMC RECEIVE DONE: This bit is write only and read as 0. When written with a 1, it will issue an RMC receive done to the REM as if it had come from the RMC bus.                                                                                                                                            |
| 4     | FRRAM    | FORCE RMC_RCV buffers to Adapter Manager                                                                                                                                                                                                                                                                         |
| 3     | FRRHST   | FORCE RMC_RCV buffers to Host                                                                                                                                                                                                                                                                                    |
| 2     | FHSTAM   | FORCE HOST_RCV buffers to Adapter Manager                                                                                                                                                                                                                                                                        |

| Bit # | Mnemonic | Definition                                                                                                                                                        |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1     | REMFZ    | REM FREEZE: When asserted will cause the REM to complete a current packet move and/or free buffer move and stall. REM will resume upon deassertion of this signal |
| 0     | REMENA   | REM Enable, ie enable the state machines that run the REM. REM parameters CAN ONLY BE WRITTEN when REMENA is deasserted                                           |

### 8.2.2 PMC RAM PARITY ERROR REGISTER (CSR01)

AM READ/WRITE if CSR00[0] = 0, ie, for test purposes only. Otherwise:

ACCESS: ADAPTER MANAGER (SC, cleared on INIT or READ)

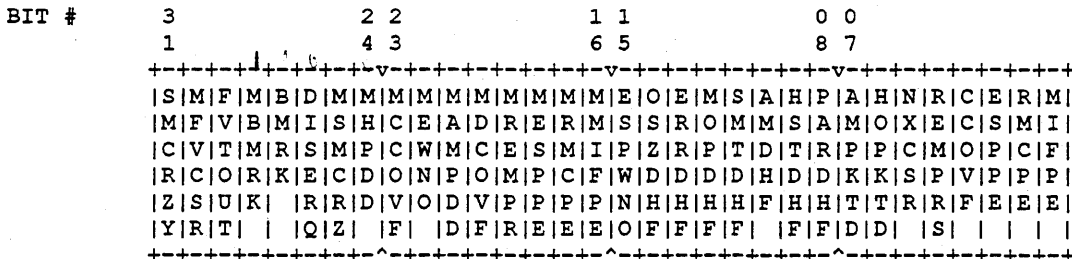


| Bit # | Mnemonic | Definition                                                                                                                                                         |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | SRAM     | If set, address is SRAM. If clear, address is DRAM.                                                                                                                |
| 30:8  | PERRAD   | Actual physical address of first detected error. Note that if a MIF parity error is not posted that this field contains the last accessed RAM read address         |
| 7     | BYT3ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 3, bits [31:24], has a parity error.     |
| 6     | BYT2ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 2, bits [23:16], has a parity error.     |
| 5     | BYT1ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 1, bits [15:8], has a parity error.      |
| 4     | BYT0ER   | If an error is in a byte protected data structure, ie, packet data or buffer descriptor, this bit set indicates that byte 0, bits [7:0], has a parity error error. |
| 3     | OWNERR   | If the error is in a PTE, then this bit set indicates that the own field, bits [35:32], has a parity error.                                                        |
| 2     | PPNERR   | If the error is in a PTE, then this bit set indicates that the PPN field, bits [31:16], has a parity error.                                                        |
| 1     | FVERR    | If the error is in a PTE, then this bit set indicates that the FV field, bits [15:0], has a parity error.                                                          |
| 0     |          | Not implemented. Read as zero.                                                                                                                                     |

### 8.2.3 PMC MASK AND INTERRUPT REGISTER (CSR02)

ACCESS: ADAPTER MANAGER (SC, cleared on INIT )

Mask Bits are R/W toggle bits, ie, write 1 to set, write 1 to clear  
 Discard Counter 1/2 full bits are read only and cleared by reading  
 the relevant counter. They can only set if CSR00[0] = 1  
 Bits 27, 7:5 are read only and cleared by reading this register  
 Bits 29, 4:0 are read only and cleared by init only

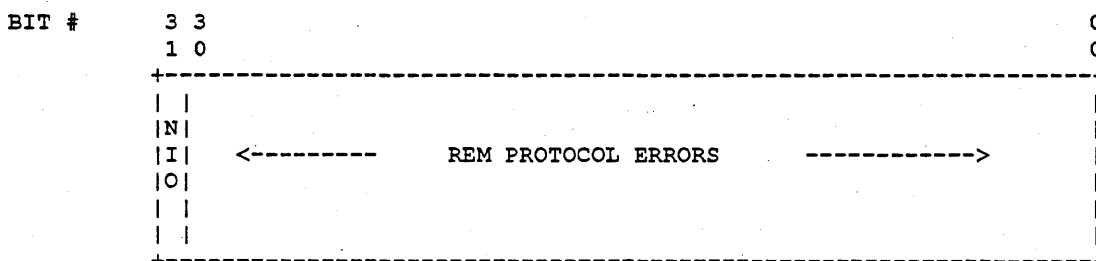


| Bit # | Mnemonic | Definition                                                                                                                                                                                                                                                  |
|-------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31    | SMCRZY   | STATE MACHINE UNKNOWN STATE : This condition causes an interrupt.                                                                                                                                                                                           |
| 30    | MFVCSR   | Mask forwarding vector timeout interrupt and access attempted to non existent CSR interrupt.                                                                                                                                                                |
| 29    | FVTOUT   | Forwarding Vector Timeout : This interrupt indicates that the PMC has waited 16 cycles since detecting an RMC_RCV clear own instruction and has not received FV Ready from the Parser Gate array.                                                           |
| 28    | MBMRK    | Mask Bookmark Interrupt.                                                                                                                                                                                                                                    |
| 27    | BMRK     | BOOKMARK : This interrupt indicates that the RMC_RCV queue has been flushed up to and including the packet current at the time of issuance of the Bookmark Request (See CSR00[16]).                                                                         |
| 26    | DISERQ   | DISABLE ESPRQENASM : This bit is readable and write 1 to set, write 1 to clear. When asserted, it will disable the affects of the ESP enable state machine. This will allow ESP requests to arbitrate for PMC memory regardless of possible RMC congestion. |
| 25    | MSMCRZ   | Mask state machine unknown state interrupt.                                                                                                                                                                                                                 |
| 24    | MHPDD    | Mask Host destined packet discarded interrupt.                                                                                                                                                                                                              |
| 23    | MCCOVF   | Mask CAS counter overflow interrupt.                                                                                                                                                                                                                        |
| 22    | MEWNO    | Mask ESP Write Not Owned interrupt.                                                                                                                                                                                                                         |
| 21    | MAMPDD   | Mask AM destined packet discarded interrupt.                                                                                                                                                                                                                |
| 20    | MDCOVF   | Mask Discard Counter Overflow Bits                                                                                                                                                                                                                          |
| 19    | MREMPR   | Mask Rem Protocol Error.                                                                                                                                                                                                                                    |
| 18    | MESPPE   | Mask ESP Parity Error.                                                                                                                                                                                                                                      |
| 17    | MRMCPE   | Mask RMC Parity Error.                                                                                                                                                                                                                                      |
| 16    | MMIFPE   | Mask MIF Protected Parity Error.                                                                                                                                                                                                                            |
| 15    | ESPWNO   | ESP Write Attempted, Not Owned. This condition causes the interrupt.                                                                                                                                                                                        |
| 14    | OSZDHF   | Oversize Discard Counter Half Full. This condition causes the interrupt.                                                                                                                                                                                    |
| 13    | ERRDHF   | Error Discard Counter Half Full. This condition causes the interrupt.                                                                                                                                                                                       |
| 12    | MOPDHF   | MOP Discard Counter Half Full. This condition causes the interrupt.                                                                                                                                                                                         |

| Bit # | Mnemonic | Definition                                                                        |
|-------|----------|-----------------------------------------------------------------------------------|
| 11    | SMTDHF   | SMT/MAC Discard Counter Half Full. This condition causes the interrupt.           |
| 10    | AMDHF    | AM Discard Counter Half Full. This condition causes the interrupt.                |
| 9     | HSTDHF   | HOST Discard Counter Half Full. This condition causes the interrupt.              |
| 8     | PARDHF   | PARSER Discard Counter Half Full. This condition causes the interrupt.            |
| 7     | AMPKTD   | AM Destined Packet Discarded. This condition causes the interrupt.                |
| 6     | HOPKTD   | HOST Destined Packet Discarded. This condition causes the interrupt.              |
| 5     | NXCSR    | Access Attempted to Nonexistent CSR Address. This condition causes the interrupt. |
| 4     | REMPRS   | REM Protocol Error Summary. This condition causes the interrupt.                  |
| 3     | CCOVF    | CAS Counter Overflowed. This condition causes the interrupt.                      |
| 2     | ESPPE    | Parity Error Detected at ESP Interface. This condition causes the interrupt.      |
| 1     | RMCPE    | Parity Error Detected at RMC Interface. This condition causes the interrupt.      |
| 0     | MIFPE    | Parity Error Detected in MIF. This condition causes the interrupt.                |

### 8.2.4 PMC PROTOCOL ERROR REGISTER (CSR03)

ACCESS: ADAPTER MANAGER (RO, cleared on INIT)



| Bit # | Mnemonic | Definition                                                                                                                        |
|-------|----------|-----------------------------------------------------------------------------------------------------------------------------------|
| 31    | NIO      | Not implemented, read as 0                                                                                                        |
| 30    | PROE30   | Protocol Error 30: Three page or more Packet. Reading intermediate Page. EOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK.        |
| 29    | PROE29   | Protocol Error 29: Three page or more Packet. Reading intermediate Page. SOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK.        |
| 28    | PROE28   | Protocol Error 28: Two page or multiple page Packet. Reading last Page. EOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK          |
| 27    | PROE27   | Protocol Error 27: Two page or multiple page Packet. Reading last Page. SOP=1. AM to HOST COPY. AM MOVE FILL LOGIC BLOCK.         |
| 26    | PROE26   | Protocol Error 26: Color Code indicates 00 (invalid) or 11 (to host). HOST to RMC or HOST to AM COPY. HOST MOVE FREE LOGIC BLOCK. |

This Register is READ ONLY and cleared only on INIT.



Digital Equipment Corporation—VAX Products and Options  
Confidential and Proprietary

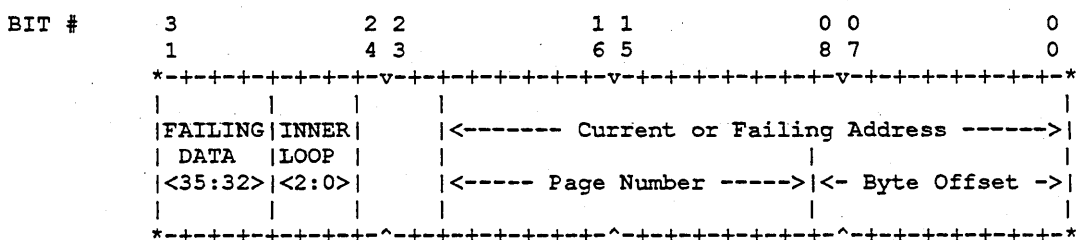
| Bit # | Mnemonic | Definition                                                                                                                                                                         |
|-------|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25    | PROE25   | Protocol Error 25: Color Code indicates 00 (invalid) or 10 (to AM). AM to RMC or AM to HOST COPY. AM MOVE FREE LOGIC BLOCK.                                                        |
| 24    | PROE24   | Protocol Error 24: Color Code indicates 00 (invalid) or 01 (to RMC). RMC to AM or RMC to HOST COPY. RMC MOVE FREE LOGIC BLOCK.                                                     |
| 23    | PROE23   | Protocol Error 23: Reading the first Page Table Entry from the HOST_RCV ring of a HOST generated Packet. HOST MOVE FILL LOGIC BLOCK.                                               |
| 22    | PROE22   | Protocol Error 22: Reading the first Page Table Entry from the AM_RCV ring of an AM generated Packet. AM MOVE FILL LOGIC BLOCK.                                                    |
| 21    | PROE21   | Protocol Error 21: Reading the first Page Table Entry from the RMC_RCV ring of a RMC generated Packet. RMC MOVE FILL LOGIC BLOCK.                                                  |
| 20    | PROE20   | Protocol Error 20: Three page or more Packet. Reading intermediate Page. EOP=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                            |
| 19    | PROE19   | Protocol Error 19: Three page or more Packet. Reading intermediate Page. SOP=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                            |
| 18    | PROE18   | Protocol Error 18: Two page or multiple page Packet. Reading last Page. EOP=0. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                             |
| 17    | PROE17   | Protocol Error 17: Two page or multiple page Packet. Reading last Page. SOP=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                             |
| 16    | PROE16   | Protocol Error 16: Two page or greater Packet. OWN=1. DISCARD S.M. RMC MOVE FILL LOGIC BLOCK.                                                                                      |
| 15    | PROE15   | Protocol Error 15: Two page or multiple page Packet. Reading last page. EOP=0. Multiple page Packet, reading intermediate page and EOP=1. AMRMCCOPY S.M. AM MOVE FILL LOGIC BLOCK. |
| 14    | PROE14   | Protocol Error 14: Two page or multiple page Packet. Reading last page. SOP=1. Multiple page Packet, reading intermediate page and SOP=1. AMRMCCOPY S.M. AM MOVE FILL LOGIC BLOCK. |
| 13    | PROE13   | Protocol Error 13: Two page or greater Packet. OWN=1. AMHOSTCOPY S.M. AM MOVE FILL LOGIC BLOCK.                                                                                    |
| 12    | PROE12   | Protocol Error 12: Two page or greater Packet. OWN=1. AMRMCCOPY S.M. AM MOVE FILL LOGIC BLOCK.                                                                                     |
| 11    | PROE11   | Protocol Error 11: Two page or greater Packet. OWN=1. HOSTAMCOPY S.M. HOST MOVE FILL LOGIC BLOCK.                                                                                  |
| 10    | PROE10   | Protocol Error 10: Two page or greater Packet. OWN=1. HOSTRMCCOPY S.M. HOST MOVE FILL LOGIC BLOCK.                                                                                 |
| 9     | PROE9    | Protocol Error 9: Two page or greater Packet. OWN=1. RMHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                                                    |
| 8     | PROE8    | Protocol Error 8: Three page or more Packet. Reading intermediate Page. EOP=1. RMHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                          |
| 7     | PROE7    | Protocol Error 7: Three page or more Packet. Reading intermediate Page. SOP=1. RMHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                          |
| 6     | PROE6    | Protocol Error 6: Two page or multiple page Packet. Reading last Page. EOP=0. RMHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                           |
| 5     | PROE5    | Protocol Error 5: Two page or multiple page Packet. Reading last Page. SOP=1. RMHOSTCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                           |
| 4     | PROE4    | Protocol Error 4: Two page or greater Packet. OWN=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                                                     |
| 3     | PROE3    | Protocol Error 3: Three page or more Packet. Reading intermediate Page. EOP=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.                                                           |

| Bit # | Mnemonic | Definition                                                                                                               |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------|
| 2     | PROE2    | Protocol Error 2: Three page or more Packet. Reading intermediate Page. SOP=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK. |
| 1     | PROE1    | Protocol Error 1: Two page or multiple page Packet. Reading last Page. EOP=0. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.  |
| 0     | PROE0    | Protocol Error 0: Two page or multiple page Packet. Reading last Page. SOP=1. RMCAMCOPY S.M. RMC MOVE FILL LOGIC BLOCK.  |

### 8.2.5 PMC PMT FAILING ADDRESS REGISTER (CSR07)

ACCESS: AM Read/Write (IFF CSROF[0] = 0 else read only) - Cleared on INIT

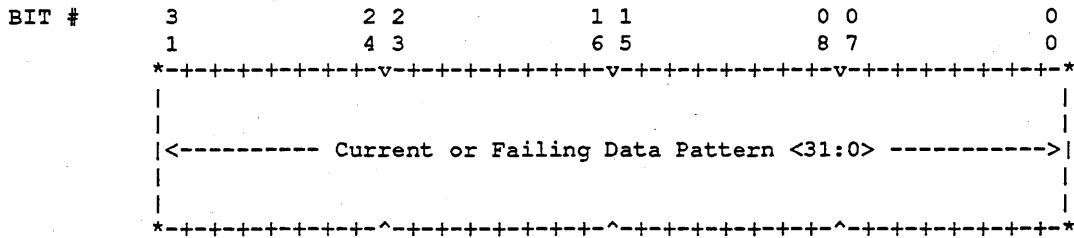
Bits 24:22 have no function. Read as 0 if CSROF[0] = 1



| Bit #   | Mnemonic                   | Definition                                                                                                                                                                                       |
|---------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <31:28> | Failing Data <35:32>       | If PMT is active, this field contains the current data pattern or, if PMT has failed, the failing pattern or if passed, the last used data pattern                                               |
| <27:25> | Inner Loop <2:0>           | This counter indicates which pattern set is current. For example if this counter = 2 it would indicate the current pattern being used for test started with the contents of PMT Data Register 2. |
| <24:22> |                            | This bits have no function. They are read/writeable for test purposes only                                                                                                                       |
| <21:0>  | Current or Failing Address | If PMT is active, this field contains the current address or, if PMT has completed, the last accessed address.                                                                                   |

### 8.2.6 PMC PMT FAILING DATA REGISTER (CSR0B)

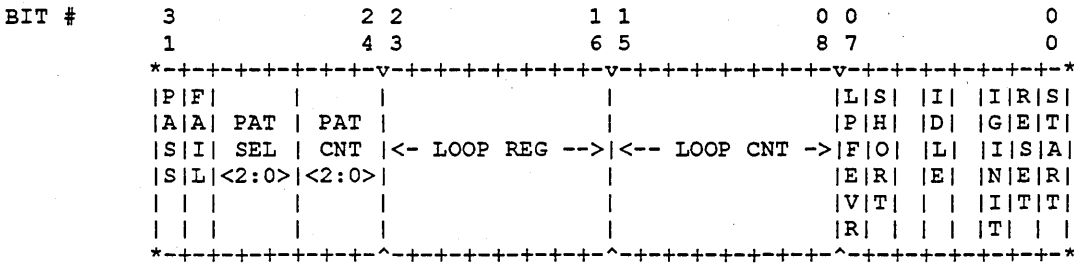
ACCESS: AM Read/Write (IFF CSROF[0] = 0 else read only) - Cleared on INIT



| Bit #  | Mnemonic                       | Definition                                                                                                                                         |
|--------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| <31:0> | Current or Failing Data <31:0> | If PMT is active, this field contains the current data pattern or, if PMT has failed, the failing pattern or if passed, the last used data pattern |

### 8.2.7 PMC PMT CONTROL AND STATUS REGISTER (CSR0F)

ACCESS: SC - INITIALIZED to ^X04000011 except bit 2 (see definition)



| Bit #   | Mnemonic      | Definition                                                                                                                                                                          |
|---------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <31>    | PASS          | When set this bit indicates that PMT was successful in testing the PMC DRAM. This bit is read only                                                                                  |
| <30>    | FAIL          | When set this bit indicates that PMT was NOT successful in testing the PMC DRAM. For further information refer to the PMT Failing address and Data registers. This bit is read only |
| <29:27> | PAT SEL <2:0> | This data field indicates which PMT data register is currently selected and being used. This field is read only                                                                     |

| Bit #   | Mnemonic       | Definition                                                                                                                                                                                                                                                                                                                                                                                   |
|---------|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <26:24> | PAT CNT <2:0>  | This data field indicates how many patterns will be used for each loop of PMT where 0 = one pattern, 1 = two patterns and so on. This field is read/write where the bits are toggle bits, ie write 1 to set, write 1 to clear if CSRF[0] = 0 else read only.                                                                                                                                 |
| <23:16> | LOOP REG <7:0> | This data field indicates which loop of PMT is currently being executed. This field is read only                                                                                                                                                                                                                                                                                             |
| <15:8>  | LOOP CNT <7:0> | This data field indicates how many loops will be used for PMT where 0 = one loop, 1 = two loops and so on. This field is read/write where the bits are toggle bits, ie write 1 to set, write 1 to clear if CSRF[0] = 0 else read only                                                                                                                                                        |
| <7>     | LPFEVR         | When set this bit causes PMT to run continuously. Such a test can be cleanly ended by resetting this bit and waiting for the normal end, ie LOOP CNT = LOOP REG. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable                                                                                                                         |
| <6>     | SHORT          | When set this bit causes PMT to apply patterns to only the first and last 8 longwords of memory for each pattern requested for test. This feature offers a fast way to stop PMT once started without initialization. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable                                                                     |
| <5>     |                | NID                                                                                                                                                                                                                                                                                                                                                                                          |
| <4>     | IDLE           | This bit when asserted indicated that PMT is not active. This bit is read only                                                                                                                                                                                                                                                                                                               |
| <3>     |                | NID                                                                                                                                                                                                                                                                                                                                                                                          |
| <2>     | IGINIT         | This bit when asserted will prevent PMT from starting in the event of a programmed init (ie a PMC init of 1 cycle). This bit is cleared on a power-up thus not preventing PMT from starting under such conditions. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable                                                                       |
| <1>     | RESET          | In the event of PMT failing, writing a 1 to this bit will allow the PMT state machine to go to its idle state. Note that PMT failing address and data registers should be read prior to setting this bit. This bit is write only and read as 0                                                                                                                                               |
| <0>     | START          | This bit when transitioned from deasserted to asserted will start PMT (if IGINIT = 0). This bit is set on initialization and is cleared by the PMT state machine upon completion of PMT. Note that deasserting this bit once PMT has started, ie, IDLE is deasserted, will not stop PMT. This bit is a toggle bit, ie write 1 to set, write 1 to clear and is unconditionally read/writeable |

## CHAPTER 9

### THE PACKET MEMORY (DRAM) TEST

#### 9.1 PACKET MEMORY TEST

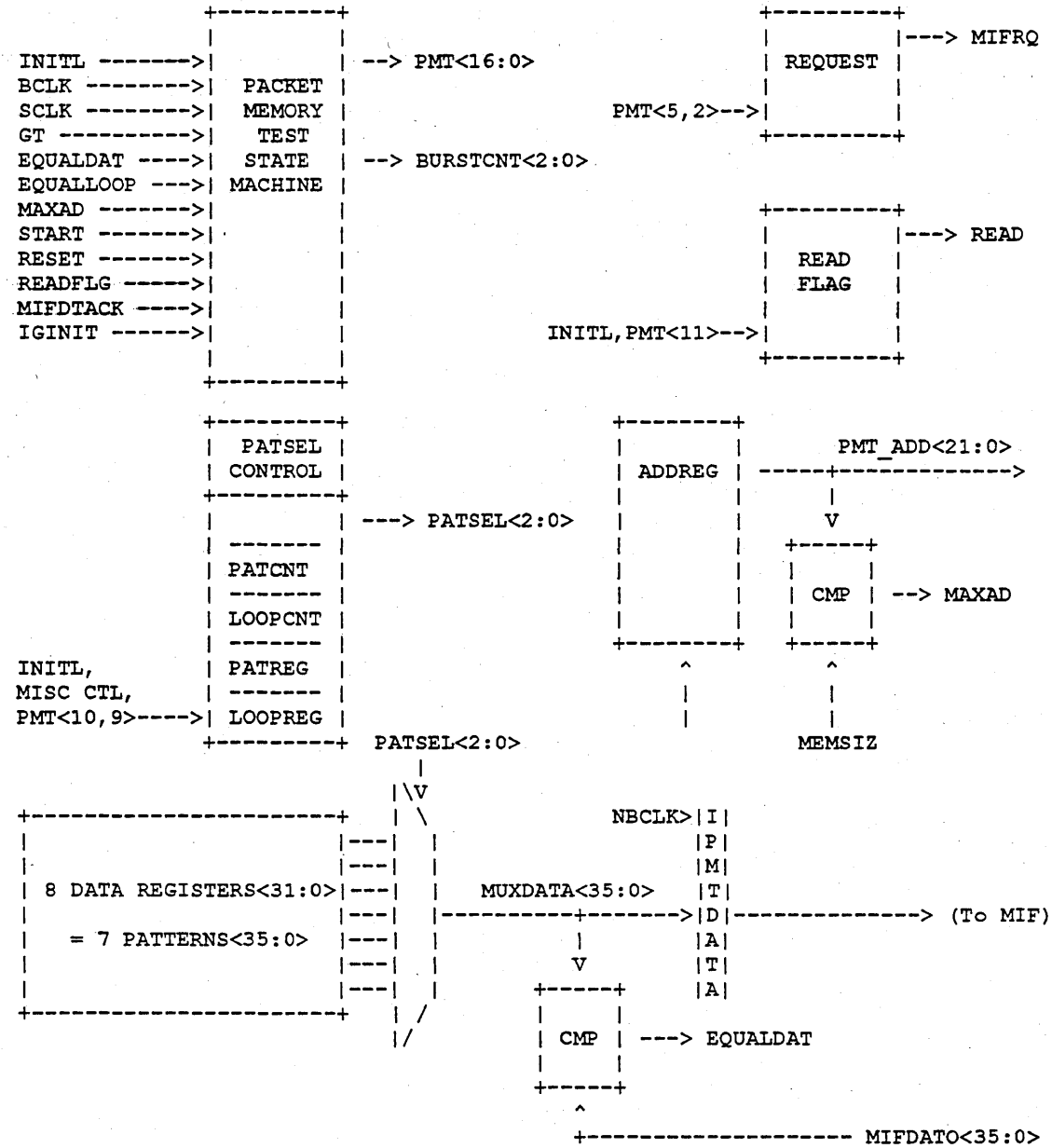
The Packet Memory Test Block tests the DRAM of the DEMFA module automatically on power up and optionally on programmed initialization or by explicit command. A number of preset (on power up) or programmed values are written to all of DRAM and then read back for verification. The pattern set may then be rotated and the process repeated (assuming more than 1 pattern selected). For non-power-up purposes, the test may be programmed to loop up to 256 times or infinitely with 1 to 7 patterns selected.

An understanding of the interface may be gleaned by referring to the descriptions of the PMT registers in the CSR REGISTERS Chapter. CSR0F is the prime PMT register; all control is exercised here. CSR07 and CSR0B provide address and data status information and CSR's 04, 08, 0C, 15, 05, 09, 0D and 17 are the PMT Data registers 0 through 7 respectively.

It should be noted that the test uses 36-bit data patterns. PMT Data register 7 contains all of the data patterns bits 35 through 32 (see definition in the CSR chapter). This allows up to 7 data patterns to be used for any given test.

One test loop is considered to be writing and reading back all of DRAM with all of the selected patterns so, for example, if 4 patterns are selected (PATSEL=3) then all of DRAM will be written/read 4 times, each time starting with a different pattern.

Figure 9-1: PMT BLOCK DIAGRAM



**APPENDIX A**  
**RING CONFIGURATIONS**

**Digital Equipment Corporation—VAX Products and Options**  
**Confidential and Proprietary**

The PTEs that compose the XMT and RCV RINGS for the AM, HOST and RMC, have been substituted with either PPN# (Physical Page Number) or UNAS (Unassigned Space). PPNs are allocated to the segments of the RING which actually have valid Physical Page Numbers. UNAS is designated to the segments of the RING which do not have Valid Physical Page Numbers.

The OWN bit is CLEAR when the REM has ownership of the RING and it is SET when the external interface has Ownership.

Pointers move in an upward direction.

The external interface pointers are preceded by EXT.

| RMC_RCV_RING        | AM_RCV_RING        | HOST_RCV_RING         |
|---------------------|--------------------|-----------------------|
| +-----+             | +-----+            | +-----+               |
| UNAS  0             | UNAS  0            | UNAS  0               |
| +-----+             | +-----+            | +-----+               |
| UNAS  0             | UNAS  0            | UNAS  0               |
| +-----+             | +-----+            | +-----+               |
| UNAS  0  <--RMC_RCV | UNAS  0  <--AM_RCV | UNAS  0  <--HOST_RCV  |
| +-----+             | +-----+            | +-----+               |
| FREE                | FREE               | FREE                  |
| +-----+             | +-----+            | +-----+               |
| PPN5  1             | PPN10  1           | PPN15  1              |
| +-----+             | +-----+            | +-----+               |
| PPN4  1             | PPN9  1            | PPN14  1              |
| +-----+             | +-----+            | +-----+               |
| PPN3  1             | PPN8  1            | PPN13  1              |
| +-----+             | +-----+            | +-----+               |
| PPN2  1             | PPN7  1            | PPN12  1              |
| +-----+             | +-----+            | +-----+               |
| PPN1  1  <--RMC_RCV | PPN6  1  <--AM_RCV | PPN11  1  <--HOST_RCV |
| +-----+             | +-----+            | +-----+               |
| FILL                | FILL               | FILL                  |
| EXT_RMC             | EXT_AM             | EXT_HOST              |
| PNTR                | PNTR               | PNTR                  |

| RMC_XMT_RING        | AM_XMT_RING        | HOST_XMT_RING        |
|---------------------|--------------------|----------------------|
| +-----+             | +-----+            | +-----+              |
| UNAS  0             | UNAS  0            | UNAS  0              |
| +-----+             | +-----+            | +-----+              |
| UNAS  0             | UNAS  0            | UNAS  0              |
| +-----+             | +-----+            | +-----+              |
| UNAS  0             | UNAS  0            | UNAS  0              |
| +-----+             | +-----+            | +-----+              |
| UNAS  0             | UNAS  0            | UNAS  0              |
| +-----+             | +-----+            | +-----+              |
| UNAS  0             | UNAS  0            | UNAS  0              |
| +-----+             | +-----+            | +-----+              |
| UNAS  0  <--RMC_XMT | UNAS  0  <--AM_XMT | UNAS  0  <--HOST_XMT |
| +-----+             | +-----+            | +-----+              |
| FILL                | FILL               | FILL                 |
| RMC_XMT             | AM_XMT             | HOST_XMT             |
| FREE                | FREE               | FREE                 |
| EXT_RMC             | EXT_RMC            | EXT_HOST             |
| PNTR                | PNTR               | PNTR                 |

rings\_at\_init.line\_art



MOVE FILL PROCESS

The changing configuration of the RINGS during a MOVE\_FILL PROCESS is illustrated below. The RMC\_RCV\_RING shows that the RMC has written an AM and a HOST Packet(s) into the PBM. The Packet(s) are represented by PPN#\A and PPN#\H. The REMs objective is to move the PTEs (PPNs) of the AM and HOST Packets to their respective AM\_XMT and HOST\_XMT RINGS in the Page Table. Top B illustrates the configuration of the RING after the first PTE of the AM Packet has been moved to the AM\_XMT RING. Top C indicates that the second PTE has been moved.

|                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                              |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> RMC_RCV_RING +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0 &lt;--RMC_RCV +-----+  FREE   PPN5  1 &lt;--EXT_RMC +-----+  PNTR   PPN4/H  0  +-----+   PPN3/H  0  +-----+   PPN2/A  0  +-----+   PPN1/A  0 &lt;--RMC_RCV +-----+  FILL Top A </pre> | <pre> RMC_RCV_RING +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0 &lt;--RMC_RCV +-----+  FREE   PPN5  1 &lt;--EXT_RMC +-----+  PNTR   PPN4/H  0  +-----+   PPN3/H  0  +-----+   PPN2/A  0 &lt;--RMC_RCV +-----+  FILL   UNAS  0  +-----+   UNAS  0  +-----+ Top B </pre> | <pre> RMC_RCV_RING +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0 &lt;--RMC_RCV +-----+  FREE   PPN5  1 &lt;--EXT_RMC +-----+  PNTR   PPN4/H  0  +-----+   PPN3/H  0 &lt;--RMC_RCV +-----+  FILL   UNAS  0  +-----+   UNAS  0  +-----+ Top C </pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

The RINGS below illustrate the filling of the AM\_XMT\_RING. The first AM PTE/PPN# is moved from the RMC\_RCV\_RING (Top B) and written into the AM\_XMT\_RING (Bottom A). Note that the Ownership bit of PPN1 remains CLEAR initially. The final PPN is now written into the AM\_XMT\_RING (See Top C and Bottom B). Note that the Ownership bit is SET for PPN2 in Bottom B. In Bottom C, the Own bit is now SET for PPN1. The /R appended to the PPNs designates the ring of origin of the PPN.

|                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                           |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> AM_XMT_RING +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0   ----\ +-----+  ----/   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0   AM_XMT +-----+  FILL   UNAS  0 &lt;--EXT_AM +-----+  PNTR   PPN1/R  0 &lt;--AM_XMT +-----+  FREE Bottom A </pre> | <pre> AM_XMT_RING +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0   ----\ +-----+  ----/   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0 &lt;--AM_XMT +-----+  FILL   PPN2/R  1 &lt;--EXT_AM +-----+  PNTR   PPN1/R  0 &lt;--AM_XMT +-----+  FREE Bottom B </pre> | <pre> AM_XMT_RING +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0  +-----+   UNAS  0 &lt;--AM_XMT +-----+  FILL   PPN2/R  1  +-----+  AM_XMT   PPN1/R  1 &lt;--FREE +-----+  EXT_AM +-----+  PNTR Bottom C </pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

move\_fill\_rings





RING CONFIGURATION DURING A MOVE\_FREE PROCESS

The RING configurations here illustrate the PTE by PTE process of emptying the RMC\_XMT RINGS. This is a continuation of the MOVE\_FREE PROCESS started on the previous page. Top A shows that PPN12 has been returned to the HOST\_RCV RING shown in Bottom A. PPN9 has been FREED from Top B and returned to the AM\_RCV RING shown in Bottom B. Top C indicates that PPN11 has been returned to the HOST\_RCV RING. The RMC\_XMT FREE Pointer in Top C indicates that the Packet represented in PBM by PPN15, has not been read by the RMC. Therefore, the MOVE\_FREE PROCESS for the RMC\_XMT RING will stop at this point.

| RMC_XMT_RING         | RMC_XMT_RING         | RMC_XMT_RING         |
|----------------------|----------------------|----------------------|
| +-----+              | +-----+              | +-----+              |
| PPN15/H 1 <--EXT RMC | PPN15/H 1 <--EXT RMC | PPN15/H 1 <--RMC_XMT |
| +-----+ PNTR         | +-----+ PNTR         | +-----+ FREE         |
| PPN11/H 0            | PPN11/H 0 <--RMC_XMT | UNAS  0  EXT_RMC     |
| +-----+              | +-----+ FREE         | +-----+ PNTR         |
| PPN9/A  0 <--RMC_XMT | UNAS  0              | UNAS  0              |
| +-----+ FREE         | +-----+              | +-----+              |
| UNAS  0              | UNAS  0              | UNAS  0              |
| +-----+              | +-----+              | +-----+              |
| UNAS  0              | UNAS  0              | UNAS  0              |
| +-----+ ----\        | +-----+ ----\        | +-----+              |
| UNAS  0  ----/       | UNAS  0  ----/       | UNAS  0              |
| +-----+              | +-----+              | +-----+              |
| UNAS  0              | UNAS  0              | UNAS  0              |
| +-----+              | +-----+              | +-----+              |
| UNAS  0 <--RMC_XMT   | UNAS  0 <--RMC_XMT   | UNAS  0 <--RMC_XMT   |
| +-----+ FILL         | +-----+ FILL         | +-----+ FILL         |
| Top A                | Top B                | Top C                |

The circular nature of the FIFOs is illustrated in Bottom C. The HOST\_RCV\_FREE Pointer has flipped to the bottom of the ring to position it underneath the HOST\_RCV\_FILL Pointer.

| HOST_RCV_RING        | AM_RCV_RING        | HOST_RCV_RING        |
|----------------------|--------------------|----------------------|
| +-----+              | +-----+            | +-----+              |
| UNAS  0 <--HOST_RCV  | UNAS  0 <--AM_RCV  | PPN11  1             |
| +-----+ FREE         | +-----+ FREE       | +-----+              |
| PPN12  1             | PPN9  1            | PPN12  1             |
| +-----+              | +-----+            | +-----+              |
| PPN14  1             | PPN8  1            | PPN14  1             |
| +-----+              | +-----+            | +-----+              |
| PPN13  1 <--HOST_RCV | PPN6  1            | PPN13  1 <--HOST_RCV |
| +-----+ FILL         | +-----+            | +-----+ FILL         |
| UNAS  0              | PPN7  1            | UNAS  0              |
| +-----+              | +-----+            | +-----+              |
| UNAS  0              | PPN10  1 <--AM_RCV | UNAS  0              |
| +-----+              | +-----+ FILL       | +-----+              |
| UNAS  0              | UNAS  0            | UNAS  0              |
| +-----+              | +-----+            | +-----+              |
| UNAS  0              | UNAS  0            | UNAS  0 <--HOST_RCV  |
| +-----+              | +-----+            | +-----+ FREE         |
| Bottom A             | Bottom C           | Bottom A             |

move\_free\_cmpd2

DYNAMIC RING CONFIGURATIONS CAUSED BY MOVE\_FILL AND MOVE\_FREE

This example shows a ring (Top A) at initialization. Top B shows the PPN for a Packet written into PBM destined for the AM. Top C shows that a second Packet of two pages has been written into the PBM (Packet Buffer Memory) and it is represented in the RMC\_RCV RING as PPN2 and PPN3. The /H indicates that the Packets DESTINATION is the HOST.

| RMC_RCV_RING       | RMC_RCV_RING         | RMC_RCV_RING         |
|--------------------|----------------------|----------------------|
| +-----++           | +-----++             | +-----++             |
| UNAS  0            | UNAS  0              | UNAS  0              |
| +-----++           | +-----++             | +-----++             |
| UNAS  0  -----\    | UNAS  0  -----\      | UNAS  0  -----\      |
| +-----++ -----/    | +-----++ -----/      | +-----++ -----/      |
| UNAS  0            | UNAS  0              | UNAS  0              |
| +-----++           | +-----++             | +-----++             |
| UNAS  0 <--RMC_RCV | UNAS  0 <--RMC_RCV   | UNAS  0 <--RMC_RCV   |
| +-----++ FREE      | +-----++ FREE        | +-----++ FREE        |
| PPN4  1            | PPN4  1              | PPN4  1 <--EXT_RMC   |
| +-----++           | +-----++             | +-----++ PNTR        |
| PPN3  1            | PPN3  1              | PPN3/H  0            |
| +-----++           | +-----++             | +-----++             |
| PPN2  1            | PPN2  1 <--EXT_RMC   | PPN2/H  0            |
| +-----++ RMC_RCV   | +-----++ PNTR        | +-----++             |
| PPN1  1 <--FILL    | PPN1/A  0 <--RMC_RCV | PPN1/A  0 <--RMC_RCV |
| +-----++ EXT_RMC   | +-----++ FILL        | +-----++ FILL        |
| Top A              | Top B                | Top C                |

The RING configuration in Bottom A shows that the PTEs were moved to the AM and HOST XMT RINGS and then the PTEs were returned to the FREE section of the RCV RING of ORIGIN. Bottom B indicates that three more Packets have been written into the PBM by the RMC. They are represented by PPN4, PPN2, AND PPN3 and CLEAR ownership. The configuration in Bottom C indicates that the PTEs in Bottom B, representing the three Packets, were moved to the appropriate XMT RINGS and then the PTEs were returned to the FREE section of the RMC\_RCV RING which was the ring that the PTEs originated from. The Ownership was SET for PTEs that were returned to FREE SPACE.

| RMC_RCV_RING       | RMC_RCV_RING         | RMC_RCV_RING       |
|--------------------|----------------------|--------------------|
| +-----++           | +-----++             | +-----++           |
| UNAS  0 <--RMC_RCV | UNAS  0 <--RMC_RCV   | PPN4  1            |
| +-----++ FREE      | +-----++ FREE        | +-----++           |
| PPN1  1            | PPN1  1 <--EXT_RMC   | PPN1  1 <--RMC_RCV |
| +-----++           | +-----++ PNTR        | +-----++ FILL      |
| PPN3  1  -----\    | PPN3/A  0            | UNAS  0  EXT_RMC   |
| +-----++ -----/    | +-----++             | +-----++ PNTR      |
| PPN2  1            | PPN2/H  0            | UNAS  0            |
| +-----++           | +-----++             | +-----++ -----\    |
| PPN4  1 <--RMC_RCV | PPN4/A  0 <--RMC_RCV | UNAS  0  -----/    |
| +-----++ FILL      | +-----++ FILL        | +-----++           |
| UNAS  0  EXT_RMC   | UNAS  0              | UNAS  0 <--RMC_RCV |
| +-----++ PNTR      | +-----++             | +-----++ FREE      |
| UNAS  0            | UNAS  0              | PPN2  1            |
| +-----++           | +-----++             | +-----++           |
| UNAS  0            | UNAS  0              | PPN3  1            |
| +-----++           | +-----++             | +-----++           |
| Bottom A           | Bottom B             | Bottom C           |

dynamic\_rcv\_rings

DISCARDING A PACKET

The ring represented in Top A indicates that two Packets have been written into the Packet Buffer Memory and are represented in the RMC\_RCV RING in the Page Table below as PPN1/A, PPN2/H and PPN3/H. In this example, the Packet is discarded for one of the two following reasons: a) Discard bit was set in the Forwarding Vector or b) Insufficient Allocation.

```
RMC_RCV_RING
+-----+
| UNAS  |0|
+-----+
| UNAS  |0|
+-----+
| UNAS  |0|<--RMC_RCV
+-----+      FREE
| PPN5  |1|
+-----+
| PPN4  |1|<--EXT_RMC
+-----+      PNTR
| PPN3/H|0|
+-----+
| PPN2/H|0|
+-----+
| PPN1/A|0|<--RMC_RCV
+-----+      FILL
Top A
```

In the RINGS below, the discarding of a Packet is illustrated. In Bottom A, PPN1/A is shown to be moved from the address in the RMC\_RCV\_FILL Pointer to the address in the RMC\_RCV\_FREE Pointer. The /A is dropped because the PPN no longer represents a Packet in PBM. The ownership is set so that this PPN may represent another Packet written into PBM. The PPN1/A is replaced with UNAS because this address is now part of RMC\_RCV\_FREE space. Bottom B and C illustrated the same procedure for the second Packet written into the RMC\_RCV RING.

```
RMC_RCV_RING
+-----+
| UNAS  |0|
+-----+
| UNAS  |0|<--RMC_RCV
+-----+      FREE
| PPN1  |1|
+-----+
| PPN5  |1|
+-----+
| PPN4  |1|<--EXT_RMC
+-----+      PNTR
| PPN3/H|1|
+-----+
| PPN2/H|1|<--RMC_RCV
+-----+      FILL
| UNAS  |0|
+-----+
Bottom A
```

```
RMC_RCV_RING
+-----+
| UNAS  |0|<--RMC_RCV
+-----+      FREE
| PPN2  |1|
+-----+
| PPN1  |1|
+-----+
| PPN5  |1|
+-----+
| PPN4  |1|<--EXT_RMC
+-----+      PNTR
| PPN3/H|1|<--RMC_RCV
+-----+      FILL
| UNAS  |0|
+-----+
| UNAS  |0|
+-----+
Bottom B
```

```
RMC_RCV_RING
+-----+
| PPN3  |1|
+-----+
| PPN2  |1|
+-----+
| PPN1  |1|
+-----+
| PPN5  |1|
+-----+
| PPN4  |1|<--EXT_RMC
+-----+      PNTR
| UNAS  |0| RMC_RCV
+-----+      FILL
| UNAS  |0|
+-----+
| UNAS  |0|<--RMC_RCV
+-----+      FREE
Bottom C
discard_pkt.diag
```