EK-DMB32-UG-001

DMB32

User Guide

digital™

EK-DMB32-UG-001

DMB32

User Guide

Prepared by Educational Services
of
Digital Equipment Corporation

First Edition, May 1986

Copyright © 1986 by Digital Equipment Corporation

All Rights Reserved

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 appear in this document.

Using Digital's networked computer systems, this book was produced electronically by the Media, Publishing and Design Services department in Reading, England.

Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation.

digital™

DEC	PDP	RT
DECmate	P/OS	UNIBUS
DECUS	Professional	VAX
DECwriter	Rainbow	VMS
DIBOL	RSTS	VT
MASSBUS	RSX	Work Processor

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 SCOPE 1-1
1.2 OVERVIEW 1-1
1.3 FUNCTIONAL DESCRIPTION 1-2
1.3.1 Data Transfer 1-4
1.3.2 Interrupts 1-6
1.3.3 Device Registers 1-6
1.3.3.1 VAXBI Registers 1-6
1.3.3.2 DMB32 Registers 1-6
1.4 PHYSICAL DESCRIPTION 1-7
1.4.1 The DMB32 Option 1-7
1.4.2 The T1012 Module 1-8
1.4.3 The H3033 Distribution Panel 1-9
1.4.4 Communications Interfaces 1-9
1.4.4.1 Asynchronous Interface 1-10
1.4.4.2 The Synchronous Interface 1-12
1.4.4.3 The Printer Interface 1-13
1.5 SPECIFICATIONS 1-13
1.5.1 Electrical Requirements 1-13
1.5.2 Functional Parameters 1-13
1.5.2.1 Synchronous Functional Parameters 1-13
1.5.2.2 Asynchronous Functional Parameters 1-14
1.5.2.3 Printer Port Functional Parameters 1-14
1.5.2.4 Throughput 1-14
1.5.3 Environmental Specifications 1-15
1.5.3.1 Operating Environment 1-15
1.5.3.2 Storage Environment 1-15

CHAPTER 2 INSTALLATION

2.1 SCOPE 2-1
2.2 INSTALLATION TASK LIST 2-1
2.3 SITE PLANNING 2-2
2.4 DMB32 INSTALLATION KITS 2-2
2.5 INSTALLATION CHECKS 2-3
2.6 CONFIGURATION RULES 2-3
2.7 MECHANICAL INSTALLATION 2-3
2.7.1 Electro-Static Discharge Precautions 2-3
2.7.1.1 Anti-Static Wrist Strap 2-3
2.7.1.2 Conductive Module Containers 2-4
2.7.2 T1012 Module Installation 2-4
2.7.3 Transition Header Assembly Installation 2-6
2.7.4 Ribbon Cable Installation 2-6

2.7.5 Distribution Panel Installation 2-6
2.8 ACCEPTANCE TESTING 2-8
2.8.1 Power-Up And Self-Test 2-9
2.8.2 Diagnostics 2-10
2.9 CABLING 2-10
2.9.1 H3033 Distribution Panel 2-10
2.9.2 Adapter Cables 2-12
2.9.2.1 V.35 Adapter Cable 2-13
2.9.2.2 V.24 Adapter Cable 2-14
2.9.2.3 RS-422 Adapter Cable 2-15
2.9.2.4 RS-423-A Adapter Cable 2-16
2.9.3 Loopback Connectors 2-16
2.9.3.1 H3196 50-Way Balanced Loopback Connector 2-17
2.9.3.2 H3195 50-Way Unbalanced Loopback Connector 2-17
2.9.3.3 H3197 25-Way Loopback Connector (Async Channels) 2-18
2.9.3.4 H3250 34-Way Loopback Connector (V.35) 2-19
2.9.3.5 H3248 37-Way Loopback Connector (V.24) 2-20
2.9.3.6 H3198 37-Way Loopback Connector (RS-422/423) 2-21

CHAPTER 3 OPERATION AND PROGRAMMING

3.1 SCOPE 3-1
3.2 OPERATION 3-1
3.2.1 DMB32 Register Map 3-1
3.2.2 Register Access 3-4
3.3 REGISTER BIT DEFINITIONS 3-5
3.3.1 Device Type Register (DTYPE) 3-6
3.3.2 VAXBI Control and Status Register (VAXBICSR) 3-6
3.3.3 Bus Error Register (BER) 3-8
3.3.4 Error Interrupt Control Register (EINTRCSR) 3-10
3.3.5 Interrupt Destination Register (INTRDES) 3-11
3.3.6 Starting Address Register (SADR) 3-11
3.3.7 Ending Address Register (EADR) 3-12
3.3.8 User Interface Interrupt Control Register (UINTRCSR) 3-12
3.3.9 General Purpose Register 0 (GPR0) 3-13
3.3.10 Maintenance Register (MAINT) 3-14
3.3.11 Async Control And Status Register (ACSR) 3-16
3.3.12 Sync Control And Status Register (SCSR) 3-17
3.3.13 Printer Control And Status Register (PCSR) 3-17
3.3.14 Device Configuration Register (CONFIG) 3-18
3.3.15 Second Async Control And Status Register (ACSR2) 3-19
3.3.16 Second Sync Control And Status Register (SCSR2) 3-20
3.3.17 Second Printer Control And Status Register (PCSR2) 3-20
3.3.18 System Page Table Register (SPTE) 3-21
3.3.19 System Page Table Size Register (SPTS) 3-21
3.3.20 Global Page Table Register (GPTE) 3-22
3.3.21 Global Page Table Size Register (GPTS) 3-22
3.3.22 Printer Prefix/Suffix Control Register (PFIX) 3-23
3.3.23 Printer Buffer Address Register (PBUFFAD) 3-24
3.3.24 Printer Buffer Count Register (PBUFFCT) 3-24
3.3.25 Printer Control Register (PCTRL) 3-25
3.3.26 Printer Carriage Counter Register (PCAR) 3-28

3.3.27 Printer Page Size Descriptor Register (PSIZE) 3-29
3.3.28 Sync Transmit Buffer 1 Address Register (TBUFFAD1) 3-29
3.3.29 Sync Transmit Buffer 1 Count/Offset Register (TBUFFCT1) 3-30
3.3.30 Sync Receive Buffer 1 Address Register (RBUFFAD1) 3-30
3.3.31 Sync Receive Buffer 1 Count/Offset Register (RBUFFCT1) 3-31
3.3.32 Sync Transmit Buffer 1 Control Register (TLNCTRL1) 3-31
3.3.33 Sync Receive Buffer 1 Control Register (RLNCTRL1) 3-33
3.3.34 Sync Line Parameters Register 1 (LPR1) 3-35
3.3.35 Sync Line Parameters Register 2 (LPR2) 3-37
3.3.36 Sync Transmit Buffer 2 Address Register (TBUFFAD2) 3-39
3.3.37 Sync Transmit Buffer 2 Count/Offset Register (TBUFFCT2) 3-40
3.3.38 Sync Receive Buffer 2 Address Register (RBUFFAD2) 3-40
3.3.39 Sync Receive Buffer 2 Count/Offset Register (RBUFFCT2) 3-41
3.3.40 Sync Transmit Buffer 2 Control Register (TLNCTRL2) 3-41
3.3.41 Sync Receive Buffer 2 Control Register (RLNCTRL2) 3-43
3.3.42 Sync Line Parameters Register 3 (LPR3) 3-44
3.3.43 Sync Buffer Control Register (BUFCTRL) 3-45
3.3.44 Async Transmission Preempt Buffer (PREEMPT) 3-47
3.3.45 Async Transmit Buffer Address Register (TBUFFAD) 3-47
3.3.46 Async Transmit Buffer Count/Offset Register (TBUFFCT) 3-48
3.3.47 Async Line Parameters Register (LPR) 3-48
3.3.48 Async Line Control Register (LNCTRL) 3-54
3.3.49 Async Line Status Register (LSTAT) 3-56
3.3.50 Async Flow Control Characters (FLOWC) 3-57
3.3.51 Async Transmit Completion FIFO (TBUF) 3-58
3.3.52 Sync Line Completion FIFO (SBUF) 3-59
3.3.53 Async Receive Buffer (RBUF) 3-61
3.4 PROGRAMMING FEATURES 3-63
3.4.1 Initialization 3-63
3.4.2 Configuration 3-64
3.4.3 Using the FIFOs 3-64
3.4.4 Receiving via the RX FIFO 3-64
3.4.5 Preempt Transfers 3-64
3.4.6 DMA Operations 3-64
3.4.6.1 Address Translation 3-65
3.4.6.2 Transmitting Data 3-66
3.4.6.3 Receiving Data 3-66
3.4.7 Interrupt Control 3-66
3.4.8 Error Codes 3-67
3.4.8.1 No Error 3-68
3.4.8.2 DMA Error 3-68
3.4.8.3 Message Error 3-69
3.4.8.4 Last Character Incomplete 3-69
3.4.8.5 Buffer Error 3-69
3.4.8.6 Modem Error 3-69
3.4.8.7 Aborted by Host 3-69
3.4.8.8 Printer Offline 3-70
3.4.8.9 Internal Error 3-70
3.4.9 Automatic Flow-Control 3-70
3.4.9.1 IAUTO.FLOW 3-71
3.4.9.2 SNDOFF 3-72
3.4.9.3 OAUTO.FLOW 3-72

3.4.9.4 DISCARD.FLOW 3-72
3.4.9.5 Flow-Control State Diagrams 3-72
3.4.9.6 Flow-Control Characters 3-74
3.4.10 Async Modem Control 3-74
3.4.11 Sync Modem Control 3-74
3.4.12 Selecting Protocols 3-75

CHAPTER 4 TROUBLESHOOTING

4.1 SCOPE 4-1
4.2 DIAGNOSTICS 4-1
4.3 ON-BOARD SELF-TEST DIAGNOSTIC 4-2
4.3.1 Starting the Self-Test 4-3
4.3.2 Self-Test Indications and Error Codes 4-3
4.3.2.1 Self-Test Error Codes in the RX FIFO 4-4
4.3.2.2 Test Summary Register Bit Definitions 4-5
4.4 EVDAK STANDALONE DIAGNOSTIC 4-6
4.4.1 Starting EVDAK 4-8
4.4.2 Options 4-9
4.4.3 EVENT Flags 4-10
4.4.4 Sections 4-11
4.4.5 Error Messages 4-11
4.5 EVDAJ ONLINE DIAGNOSTIC (Async) 3-11
4.5.1 Starting EVDAJ 4-13
4.5.2 Options 4-14
4.5.3 EVENT Flags 4-14
4.5.4 Sections 4-14
4.5.5 Error Messages 4-15
4.6 EVDAL ONLINE DIAGNOSTIC (Sync) 4-15
4.6.1 Starting EVDAL 4-16
4.6.2 EVENT Flags 4-16
4.6.3 Sections 4-17
4.6.4 Error Messages 4-17
4.7 EVAAA ONLINE DIAGNOSTIC (Printer) 4-17
4.7.1 Starting EVAAA 4-18
4.7.2 Error Messages 4-19
4.8 USER ENVIRONMENT TEST PROGRAM (UETP) 4-19
4.9 DATA COMMUNICATIONS LINK TEST (DCLT) 4-19
4.10 FIELD REPLACEABLE UNITS (FRUs) 4-19
4.11 TROUBLESHOOTING FLOWCHART 4-21

APPENDIX A GLOSSARY

FIGURES

Figure No. **Title**
1-1 DMB32 Typical Configuration 1-3
1-2 DMB32 Functional Block Diagram 1-5
1-3 T1012 Module 1-8
1-4 H3033 Distribution Panel 1-9
1-5 J0 to J7 Async Connector Detail 1-10
1-6 Sync Connector Detail 1-12

1-7 Printer Connector Detail..... 1-13
2-1 DMB32 Installation..... 2-5
2-2 The VAXBI Backplane (Rear View)..... 2-7
2-3 The H3033 Distribution Panel (Rear View)..... 2-8
2-4 H3033 Distribution Panel (Front View)..... 2-11
2-5 50-Way Sync Channel Connector..... 2-12
2-6 BC19F-02 (17-01112-01) Adapter Cable Detail..... 2-13
2-7 BC19D-02 (17-01110-01) Adapter Cable Detail..... 2-14
2-8 BC19B-02 (17-01108-01) Adapter Cable Detail..... 2-15
2-9 BC19E-02 (17-01111-01) Adapter Cable Detail..... 2-16
2-10 H3197 Loopback Connector..... 2-18
2-11 H3250 Loopback Connector..... 2-19
2-12 H3248 Loopback Connector..... 2-20
2-13 H3198 Loopback Connector..... 2-21
3-1 DMB32 Register Map..... 3-2
3-2 Handling an Unaligned DMA Buffer..... 3-65
3-3 DMB32 Interrupt Vectors..... 3-67
3-4 Automatic Flow-Control of Transmitted Characters..... 3-72
3-5 Automatic Flow-Control of Received Characters..... 3-73
3-6 Program Initiated Flow-Control..... 3-73

TABLES

Table No. **Title**
1-1 EIA/CCITT Signal Relationships..... 1-11
1-2 Synchronous Functional Parameters..... 1-13
1-3 Asynchronous Functional Parameters..... 1-14
1-4 Printer Port Functional Parameters..... 1-14
1-5 Maximum Sensible Speeds (Sync Port)..... 1-15
2-1 DMB32 Installation Kit Details..... 2-2
2-2 Ribbon Cable Connections..... 2-6
2-3 H3196 Balanced Loopback Interconnections..... 2-17
2-4 H3195 Unbalanced Loopback Interconnections..... 2-17
3-1 DMB32 Registers..... 3-3
4-1 Scope and Duration of Self-Test..... 4-2
4-2 Self-Test Error Codes in the RX FIFO..... 4-4
4-3 EVDAK Tests..... 4-7
4-4 Ribbon Cable Signal Distribution..... 4-20

PREFACE

This document describes the installation, use, programming, and service requirements of the DMB32 asynchronous/synchronous multiplexer. It contains information for first-line service, field service support, and for customer engineers and programmers.

The manual is organized into four chapters plus an appendix.

Chapter 1	–	Introduction
Chapter 2	–	Installation
Chapter 3	–	Operation and Programming
Chapter 4	–	Troubleshooting
Appendix A	–	Glossary

The following is a list of related titles.

Document	**Number**
DMB32 Technical Description	EK-DMB32-TD
DMB32 Field Maintenance Print Set	MP-01797-01

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This chapter describes the functions, versions, physical features, and specifications of the DMB32 communications adapter.

1.2 OVERVIEW

The DMB32 is an intelligent synchronous/asynchronous multiplexer which provides eight full-duplex asynchronous serial data channels, one synchronous data channel, and one line printer interface on VAXBI systems.

The main features of the DMB32 are:

- Single VAXBI module (T1012) and distribution panel (H3033)
- Eight full-duplex asynchronous (async) data channels
- One synchronous (sync) data channel
- One line-printer port. The DMB32 printer port supports the LP32 generic printer specification. This includes the LN01, LN01-B, LN01-S, LP25, LP26, LP27, LXY12, and LXY22 printers.
- Direct Memory Access (DMA) or single-character programmed transfers to and from host memory
- Onboard virtual-to-physical address translation allows the DMB32 to handle data buffers in their virtual address format
- Large 512-entry First-In First-Out (FIFO) buffer on the async channels, for received characters, dataset status changes, and diagnostic information
- Async ports are electrically and mechanically compliant with RS-232-C, and compatible with V.28/V.24 (T1012 is also compatible with RS-423-A, V.10, and X.26, but the H3033 is not, due to pin limitations on the 25-pin D-type connectors).
- The sync port is electrically and mechanically compliant with RS-232-C, RS-422-A/RS-449, V.11, X.27, and V.35, and compatible with RS-423-A/RS-449, V.28/V.24, V.10, and X.26.
- IBM Bisync, SDLC/HDLC, and DDCMP protocols are supported on the sync port. NRZI support is also provided.
- General Byte (GEN BYTE) protocol provides basic framing and data transfer facilities to implement other byte-oriented protocols on the sync channel
- Full-duplex point-to-point or auto-answer dial-up operation

- Programmable split-speed operation
- Total module throughput 21000 characters per second
- Automatic flow-control of transmitted and received data on the async channels
- Self-test diagnostics
- Programmable loopback modes and test facilities
- There are no switches on the module or the distribution panel. Each external cable for the sync channel is electrically coded to select a specific CCITT/EIA standard. Other line characteristics are set under program control.

Enough modem control is provided on all synchronous and asynchronous channels to allow auto-answer dial-up operation over the public switched telephone network (PSTN). The DMB32 can also be used for point-to-point operation over private lines. Modem control is implemented by software in the host.

An integral microprocessor releases the host from many of the data-handling tasks.

The DMB32 has an on-board self-test diagnostic that is executed independently of the host. Online and standalone diagnostics are also available.

Two yellow LEDs on the T1012 module give the GO/NOGO status of the option. The self-test also provides more detailed error and/or status information through the RX FIFO and some of the device registers.

1.3 FUNCTIONAL DESCRIPTION

The DMB32 is a communications adapter of the VAXBI family. It has eight asynchronous channels, one synchronous channel, and a printer channel. The function of the DMB32 is to transfer data between a VAXBI node and the asynchronous, synchronous, and printer ports.

Figure 1-1 shows a DMB32 in a typical configuration.

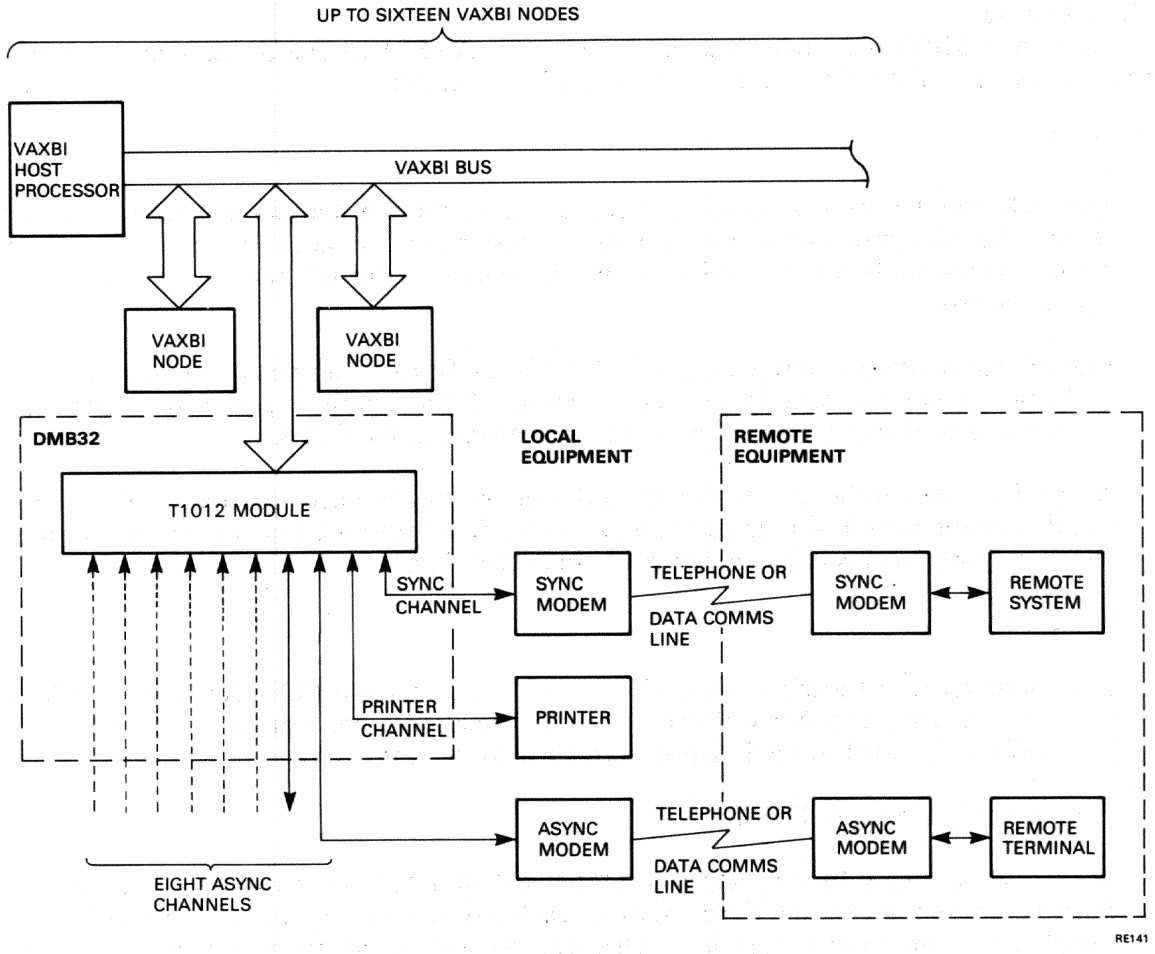


Figure 1-1 DMB32 Typical Configuration

Figure 1-2 is a functional block diagram that shows the data routes through the option. Data is transferred between a VAXBI node on the host and the communications lines via registers and DMA files in the Common Address Store RAM (CASRAM). Also in the CASRAM are Control and Status Registers (CSRs) which configure and program the option. The information which is written to and read from the CSRs is used to support and control the communications functions.

Data can be transferred between the VAXBI and CASRAM by programmed Read or Write commands, or by the DMB32's DMA logic. Data is transferred between the communications interface and the CASRAM under the control of a 68000 microprocessor.

The maximum data transfer rate on the VAXBI is 13.3 Mbytes per second. Such a high rate makes the design of the VAXBI interface critical, so a standard VAXBI interface (VAXBI corner) has been designed for all VAXBI options. The VAXBI corner is based on a specially designed VAXBI integrated circuit (BIIC), which handles all the VAXBI commands and protocols.

The intelligence of the DMB32 is supplied by the 68000 microprocessor driven by ROM-based firmware. The microprocessor controls and configures the communications interface, and manages both DMA and communications functions. The firmware also includes self-test routines which run automatically at power-up or reset.

1.3.1 Data Transfer

The function of the DMB32 is to transfer data between the VAXBI host and the data lines connected to the communications interface. There are several methods of transfer.

- DMA

  Sync RX and TX data is transferred between host memory buffers and CASRAM files by DMA. The microprocessor keeps track of the buffers, and initiates further DMA transfers as needed. Transfers between CASRAM and the sync port are initiated by an interrupt to the microprocessor.

  Printer data is also transferred across the VAXBI by DMA. Transfers of data from CASRAM to the printer port are done by the microprocessor. Printer status is polled by the microprocessor, which also makes the status information available in the CSRs.

  Async TX data can be transferred by DMA or by programmed transfer. DMA transfers to CASRAM are controlled in the same way as sync data transfers. The microprocessor monitors the async ports to check if they are ready to accept data.

- RX FIFO

  The microprocessor transfers async RX data to a 512-character RX FIFO in CASRAM. The host reads the data, together with error status information, from a single location (RBUF). The RX FIFO is also used to send diagnostic reports to the host.

- Programmed (Preempt) Transfer

  Async TX data may be transferred to CASRAM by DMA, or it can be written to a single-character 'preempt' register (one exists in each async channel). From there the transfer is completed by the microprocessor. A character written to the preempt register of a channel which is transmitting a DMA buffer will be transferred to the async port before any remaining DMA data.

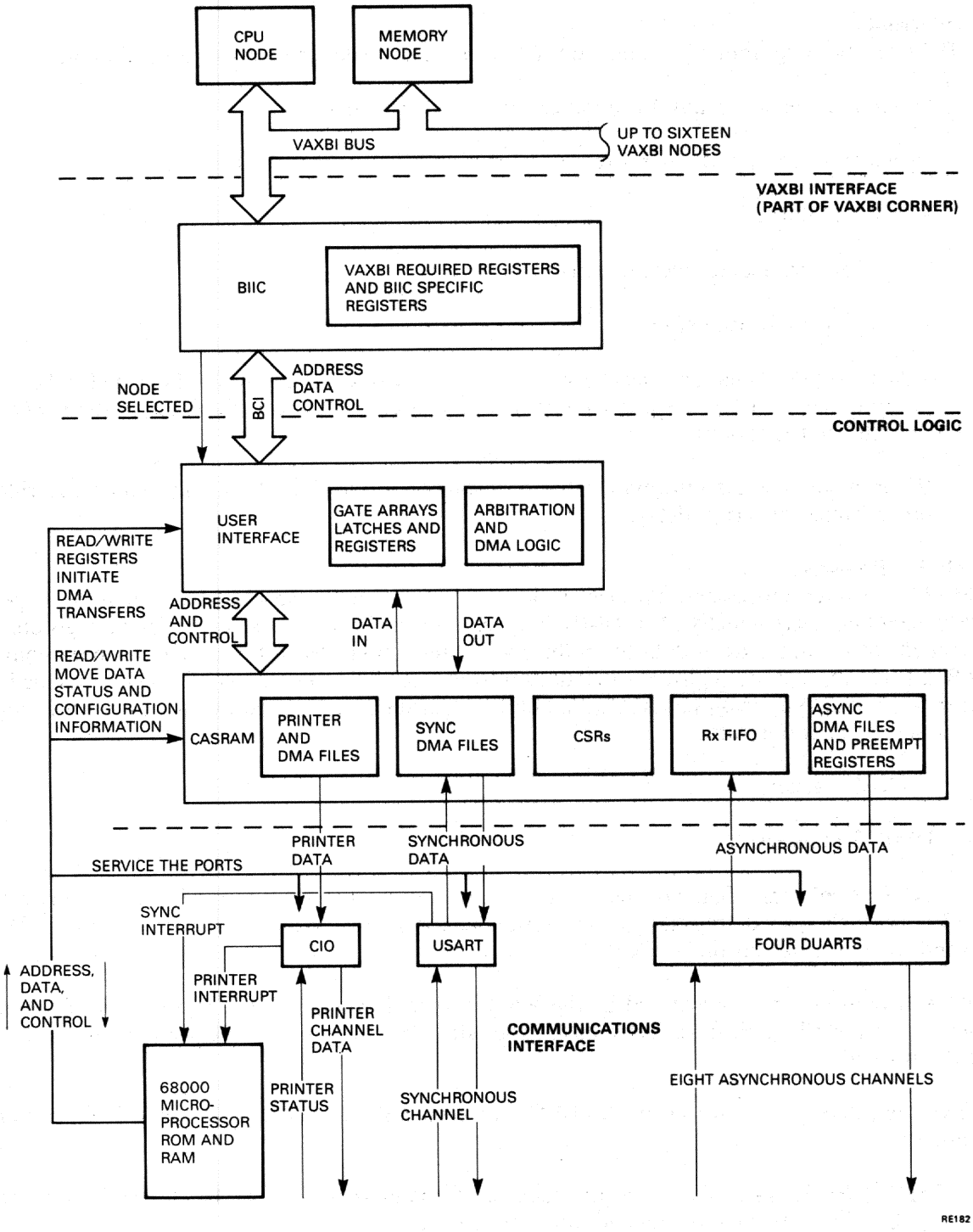


Figure 1-2 DMB32 Functional Block Diagram

1.3.2 Interrupts

The DMB32 can be programmed to interrupt a VAXBI host under the following conditions.

- When a channel is ready for another preempt character
- When transfer of a DMA buffer from system memory to an I/O channel has:
  - been aborted
  - been terminated due to an error
  - completed successfully
- When a received character has been placed in a previously empty RX FIFO (the DMB32 can be programmed to delay this interrupt so that several characters can be placed in the FIFO before the interrupt is raised)
- When modem status information has been placed in the RX FIFO. This action overrides any programmed interrupt delay.

1.3.3 Device Registers

The DMB32 is a microprocessor-controlled device and consequently the host does not have to perform many data-handling tasks directly. The DMB32 appears to the host to be a set of device registers which can be loaded with commands or data to produce a certain action. When this action has been completed the DMB32 loads appropriate registers with status information. The completion of a task is signaled to the host either by an interrupt or by status information in a register that is polled by the host. Device registers can be considered in two groups.

- VAXBI registers
- DMB32 registers

1.3.3.1 VAXBI Registers – Some of these registers must exist on any device (node) that is connected on the VAXBI. These "required" registers hold status and control information which defines how the DMB32 will respond to commands on the VAXBI.

Other VAXBI registers are not required by the VAXBI specification, but they support specific functions implemented by the BIIC. Features selected by the BIIC registers include the commands to be implemented by this node.

The VAXBI registers, which are all in the BIIC, are considered in detail in Chapter 3, Operation and Programming.

1.3.3.2 DMB32 Registers – These are the registers through which the DMB32 specific features are controlled. They can be further subdivided into four groups.

- General registers – Used to control parameters and functions common to all parts of the communications interface
- Async registers – Used to control and monitor the async ports

- Sync registers – Used to control and monitor the sync ports
- Printer registers – Used to control and monitor the printer port

A register map, together with full details of all register functions, is given in Chapter 3, Operation and Programming.

1.4 PHYSICAL DESCRIPTION

1.4.1 The DMB32 Option

The DMB32 option is made up of the following major items:

- A single VAXBI module (T1012)
- A distribution panel (H3033)
- Six ribbon cables (17-00740-xx, where xx defines the length)
- A 50-pin unbalanced loopback connector (H3195)
- A 50-pin balanced loopback connector (H3196)
- An async line loopback connector (H3197)

In addition to the above, the following adapter cables are available to match the synchronous channel connector (50-pin subminiature D-type) to the standard used by the connected equipment. Appropriate loopback connectors, not supplied with the option, are also indicated.

Cable	DIGITAL Part Number	Standard	Loopback Connector
BC19F-02	17-01112-01	V.35	H3250
BC19D-02	17-01110-01	V.24	H3248
BC19B-02	17-01108-01	RS-422	H3198
BC19E-02	17-01111-01	RS-423-A	H3198

Details of the adapter cables and all loopback connectors are given in Chapter 2, Section 2.9.

1.4.2 The T1012 Module

The DMB32 is based on a standard VAXBI-size module (T1012). The layout of this module is shown in Figure 1-3. The dimensions are 23.3 cm (9.18 in) by 20.3 cm (8.00 in).

The module is connected to the VAXBI via connector segments A and B. Connector segments C, D, and E are connected to the communications lines via the ribbon cables and the H3033 distribution panel.

The T1012 module connects directly to any slot (except number K1J1; see Chapter 2, Installation) in the VAXBI backplane. The bus address is determined by the 'Node ID' plug which is fitted to the back of the backplane (there are no switches on the module itself).

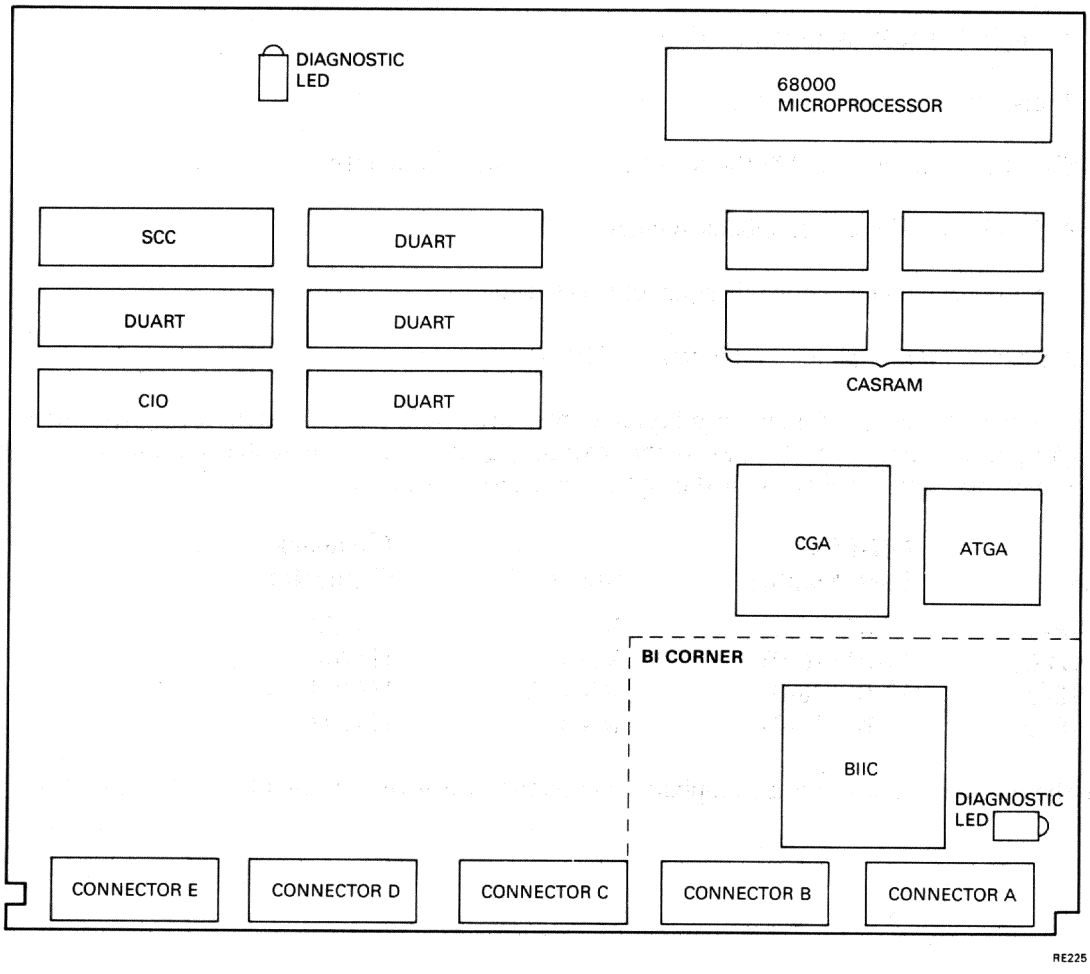


Figure 1-3 T1012 Module

1.4.3 The H3033 Distribution Panel

Figure 1-4 shows the H3033 distribution panel. This identifies the sync, async and printer connectors, and the connectors to which the ribbon cable headers are connected.

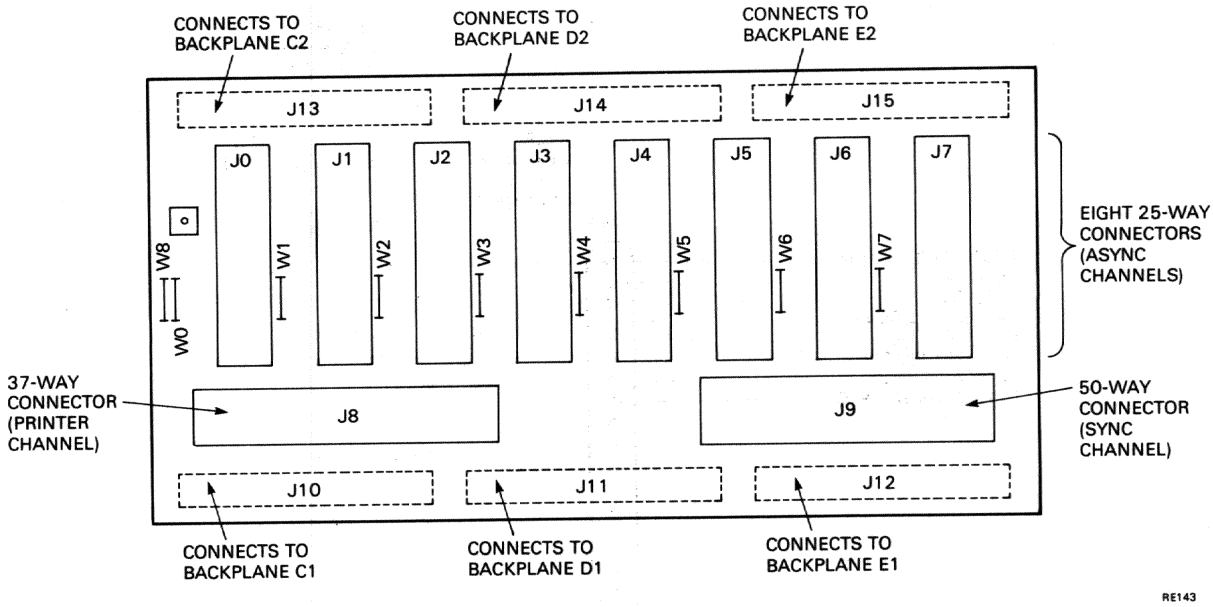


Figure 1-4 H3033 Distribution Panel

1.4.4 Communications Interfaces

Electrically and mechanically, the T1012/H3033 async ports are:

- Compliant with RS-232-C
- Compatible with V.28/V.24

The T1012 module is also compliant with RS-423-A, V.10, and X.26, but the H3033 is not, because of pin limitations on the 25-pin D-type connector.

Electrically and mechanically, the T1012/H3033 sync port is:

- Compliant with RS-232-C, RS-422-A/RS-449, V.11, X.27, and V.35
- Compatible with RS-423-A/RS-449, V.28/V.24, V.10, and X.26

1.4.4.1 Asynchronous Interface – Connection to external equipment, for each of the eight async interfaces, is made via a 25-pin subminiature D-type connector. Figure 1-5 shows the pin configuration of the connector.

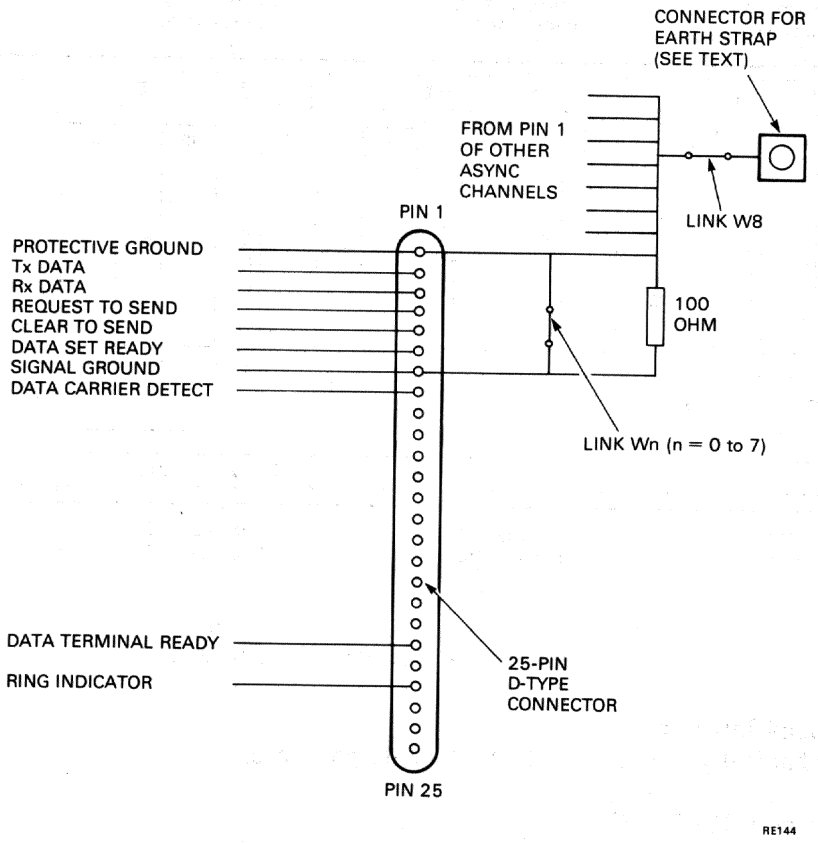


Figure 1-5 J0 to J7 Async Connector Detail

As supplied by DIGITAL, each async connector has Signal Ground (pin 7) connected to Protective Ground (pin 1) by a resistor. Provision is made on the circuit board for a wire link to connect pin 7 directly to pin 1. There are places for eight such links designated W0 to W7.

The Protective Grounds of all eight channels are joined together, and are connected (via wire link W8) to a plated hole suitable for connection to a grounding strap. This allows the Protective Grounds to be connected to the chassis of the equipment in which the DMB32 is installed.

Table 1-1 shows the EIA/CCITT signal relationships and the pin connections for equivalent signals.

Table 1-1 EIA/CCITT Signal Relationships

EIA RS-449 Signal Name		Pin	EIA RS-232-C Signal Name		Pin	CCITT V.24 Signal Name		Pin
–	Shield	1	AA	Protective Ground	1	–		–
SG	Signal Ground	19	AB	Signal Ground	7	102	Signal Ground	7
SC	Send Common	37	–		–	–		–
RC	Receive Common	20	–		–	–		–
IS	Terminal In Service	36	–		–	–		–
IC	Incoming Call	15	CE	Ring Indicator	22	125	Calling Indicator	22
TR	Terminal Ready (+)	12	CD	Data Terminal Ready	20	108/2	Data Terminal Ready	20
TR	Terminal Ready (–)	30	–		–	–		–
DM	Data Mode (+)	1	CC	Data Set Ready	6	107	Data Set Ready	6
DM	Data Mode (–)	29	–		–	–		–
SD	Send Data (+)	4	BA	Transmitted Data	2	103	Transmitted Data	2
SD	Send Data (–)	22	–		–	–		–
RD	Received Data (+)	6	BB	Received Data	3	104	Received Data	3
RD	Received Data (–)	6	–		–	–		–
TT	Terminal Timing (+)	17	DA	Transmitter Signal Element Timing (DTR Source)	24	113	Transmitter Signal Element Timing (DTR Source)	24
TT	Terminal Timing (–)	35	–		–	–		–
ST	Send Timing (+)	5	DB	Transmitter Signal Element Timing (DCE Source)	15	114	Transmitter Signal Element Timing (DCE Source)	15
ST	Terminal Timing (–)	23	–		–	–		–
RT	Receive Timing (+)	8	DD	Receiver Signal Element Timing	17	115	Receiver Signal Element Timing	17
RT	Receive Timing (–)	26	–		–	–		–
RS	Request To Send (+)	7	CA	Request To Send	4	105	Request To Send	4
RS	Request To Send (-)	25	–		–	–		–
CS	Clear To Send (+)	9	CB	Clear To Send	5	106	Clear To Send	5
CS	Clear To Send (–)	27	–		–	–		–
RR	Receiver Ready (+)	13	CF	Received Line Signal Detector	8	109	Data Channel Received Line Signal Detector	8
RR	Receiver Ready (–)	81	–		–	–		–
SQ	Signal Quality	33	CG	Signal Quality Detector	21	110	Data Signal Quality Detector	21
NS	New Signal	34	–		–	–		–
SR	Signaling Rate Selector	16	CH	Data Signal Rate Selector (DTE Source)	23	111	Data Signaling Rate Selector (DCE Source)	23
SI	Signaling Rate Indicator	2	CI	Data Signal Rate Selector (DCE Source)	23	–		–
LL	Local Loopback	10	–		–	141	Local Loopback	18
RL	Remote Loopback	14	–		–	140	Remote Loopback	21
TM	Test Mode	18	–		–	142	Test Indicator	25

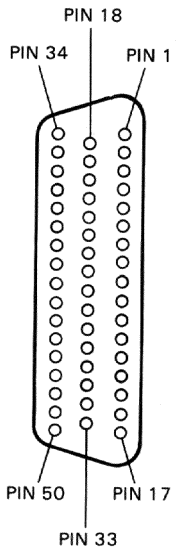
Table 1-1 EIA/CCITT Signal Relationships (continued)

EIA RS-449 Signal Name		Pin	EIA RS-232-C Signal Name	Pin	CCITT V.24 Signal Name	Pin
SS	Select Standby	32	–	–	–	–
SB	Standby Indicator	36	–	–	–	–

1.4.4.2 The Synchronous Interface – Connection to external equipment is made via a 50-pin subminiature D-type connector. Figure 1-6 shows the pin configuration of the connector. Different adapter cables are used to select only those signals needed to implement a specific interface standard. These adapter cables are fully described in Chapter 2, Section 2.8, Cabling.

50-WAY PIN	SIGNAL NAME
1	CODE GROUND
2	CODE 0
3	CODE 1
4	CODE 2
5	CODE 3
6	Tx DATA (A)
7	Tx DATA (B)
8	Tx DATA
9	RTS/C (A)
10	RTS/C (B)
11	Rx DATA (A)
12	Rx DATA (B)
13	LOCAL LOOP
14	SPEED INDICATE
15	TEST I
16	REM.LOOP
17	RI
18	Rx CLOCK (A)
19	Rx CLOCK (B)
20	Tx CLOCK (A)
21	Tx CLOCK (B)
22	CLOCK
23	V.35 Tx CLOCK (A)
24	V.35 Tx CLOCK (B)
25	V.35 CLOCK (A)
26	V.35 CLOCK (B)
27	V.35 Rx DATA (A)
28	V.35 Rx DATA (B)
29	V.35 Tx DATA (A)
30	V.35 Tx DATA (B)
31	V.35 Rx CLOCK (A)
32	V.35 Rx CLOCK (B)
33	DTR
34	DSR (A)
35	DSR (B)
36	RTS
37	DCD/I (A)
38	DCD/I (B)
39	CTS (A)
40	CTS (B)
41	DCE GROUND
42	TEST 1
43	TEST 2
44	DTE GROUND
45	DTR (A)
46	DTR (B)
47	CLOCK (A)
48	CLOCK (B)
49	TEST 3
50	SPEED

(A),(B) WIRES A(+ve) AND B(– ve) OF A TWISTED PAIR



50-WAY D-TYPE CONNECTOR (MALE PLUG)

RE146

Figure 1-6 Sync Connector Detail

1.4.4.3 The Printer Interface – The printer port is a parallel interface at TTL levels. Connection to the printer is made via a 37-pin subminiature D-type connector. Figure 1-7 shows the pin configuration of the connector.

The T1012/H3033 line printer port supports the LP32 generic printer specification. This includes the LN01, LN01-B, LN01-S, LP25, LP26, LP27, LXY12, and LXY22 printers.

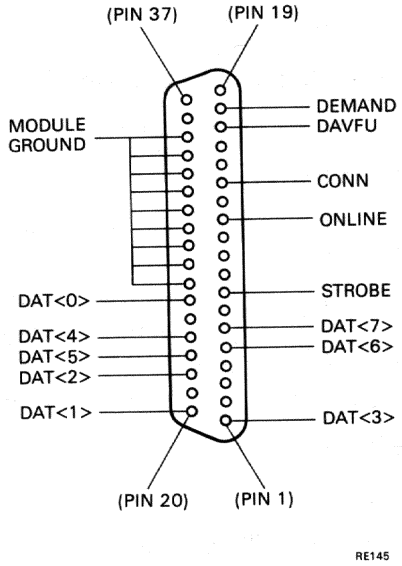


Figure 1-7 Printer Connector Detail

1.5 SPECIFICATIONS

1.5.1 Electrical Requirements

+5 V dc + or – 5% at 6.75 A

+12 V dc + or – 3% at 300 mA

–12 V dc + or – 3% at 425 mA

1.5.2 Functional Parameters

1.5.2.1 Synchronous Functional Parameters – Table 1-2 lists the functional parameters for the synchronous line of the DMB32.

Table 1-2 Synchronous Functional Parameters

Parameter	Description
DMA transfer	Double buffered
Protocols supported	DDCMP, SDLC, HDLC, IBM BISYNC, GEN BYTE
Data rates (bits/s)	600, 1200, 1800, 2000, 2400, 4800, 9600, 19200, 48000, 56000, 64000

1.5.2.2 Asynchronous Functional Parameters – Table 1-3 lists the functional parameters for the asynchronous line of the DMB32.

Table 1-3 Asynchronous Functional Parameters

Parameter	Description
Operating mode	Full-duplex or half-duplex*
Data format	One start bit; 1, 1.5, or 2 stop bits
Character size	5, 6, 7, or 8 bits (Does not include the parity bit)
Parity	Odd, even or no parity
Data rates (bits/s)	50, 75, 110, 134.5, 150, 300, 600, 1800, 2000, 2400, 4800, 7200, 9600, 19200, 38400
Split-speed	When using split-speed operation the transmit and receive speeds of both channels sharing a DUART must be in the same group (A or B). Channels are paired (use the same DUART) as follows: channels 0 and 1, 2 and 3, 4 and 5, 6 and 7. Speeds are grouped as follows:

Speeds (bits/s)	Group
50, 7200, 38400	A
75, 150, 1800 2000, 19200	B
110, 134.5, 300, 600, 1200, 2400, 4800, 9600	A and B

* Half-duplex is only supported on systems using coded link-control as the DMB32 does not support the secondary transmit and receive signals.

1.5.2.3 Printer Port Functional Parameters – Table 1-4 lists the functional parameters for the printer port of the DMB32.

Table 1-4 Printer Port Functional Parameters

Parameter	Description
Formatting capabilities	Prefix characters, suffix characters, CR insertion, FF to LF conversion, lower-case to upper-case conversion, line wrapping, carriage position tracking, line truncate.

1.5.2.4 Throughput – Each asynchronous channel is capable of full-duplex operation at data rates of up to 38400 bits/s. However, the DMB32 cannot support eight channels operating at 38400 bits/s at the same time. The total maximum throughput of the DMB32 is around 21000 characters per second (eight data bits with start and one stop bit), including all characters transmitted and received on the synchronous line and the printer port.

If congestion occurs, the DMB32 will give priority to the synchronous line, followed by the reception of async characters.

The individual maximum througputs for each port are:

- Sync – 16000 char/s
- Async – 10000 char/s
- Printer – 4000 chars/s (equivalent to 1800 lines/min)

The maximum sensible speed that can be used on the sync port depends on the protocol and electrical interface standard selected. Table 1-5 lists the maximum sensible speeds for all the supported protocols.

Table 1-5 Maximum Sensible Speeds (Sync Port)

	Electrical Interface Standard				
Protocol	**RS-232**	**RS-423**	**RS-422**	**V.35**	
DDCMP	19200	19200	19200	19200	**Data rate (Bits/s)**
HDLC	19200	64000	64000	48000	
BISYNC	9600	9600	9600	9600	
GEN BYTE	9600	9600	9600	9600	

1.5.3 Environmental Specifications

1.5.3.1 Operating Environment

- Temperature: 5°C to 50°C (41°F to 122°F)
- Relative humidity: 10% to 95% non-condensing, with a maximum wet bulb of 32°C (90°F) and a minimum dew point of 2°C (36°F)
- Altitude: Up to 2.4 km (8000 feet)

1.5.3.2 Storage Environment

- Temperature: –40°C to 66°C (–40°F to 151°F)
- Relative Humidity: Up to 95% non-condensing
- Altitude: Up to 9.0 km (30000 feet)

CHAPTER 2
INSTALLATION

2.1 SCOPE

The procedures for installing and testing the DMB32 option are described in this chapter.

WARNING

The procedures described in this chapter involve the removal of the system covers, and should be performed only by trained personnel.

CAUTION

You must wear an anti-static wrist strap connected to an active ground whenever you work on a system with the covers removed, or handle the T1012 module.

The T1012 module is supplied in protective anti-static packaging. Do not remove the module from its packaging until you are about to install it.

NOTE

The complete equipment and documentation should be present before installation begins. Any missing items must be identified and the discrepancy corrected.

2.2 INSTALLATION TASK LIST

The installation of the DMB32 consists of a number of steps:

- Unpacking and inspection (see Section 2.4)
- Installation checks (Section 2.5)
- VAXBI Configuration checks (Section 2.6)
- Installing the T1012 module (Section 2.7.2)
- Installing the transition header assembly on the VAXBI backplane (Section 2.7.3)
- Installing the ribbon cables into the transition header (Section 2.7.4)
- Installing the H3033 distribution panel and ribbon cables (Section 2.7.5)
- Installing the external adapter cables (Sections 2.7.5 and 2.9)
- Installation testing (Section 2.8)

2.3 SITE PLANNING

No special site planning is required for the DMB32 option. Your System documentation will define any site-planning considerations. The environmental conditions required by the DMB32 option are specified in Chapter 1, Section 1.5.3.

2.4 DMB32 INSTALLATION KITS

The DMB32 option is supplied in a kit that contains the parts needed to install the option, but you may also need the torque wrench (29-17381-00) from the VAXBI installation kit. Check the contents of your DMB32 installation kit against the list given in this section. Examine all parts for physical damage. Report damaged or missing items to the shipper immediately, and inform the local DIGITAL office.

CAUTION

The T1012 module is supplied in protective anti-static packaging. Do not remove the module from its packaging unless you are wearing an anti-static wrist strap.

There are three different versions of the installation kit, tailored to three different system configurations. These are:

- DMB32-LJ for the VAX-8800 and VAX-8500 systems
- DMB32-LM for the VAX-8200 and VAX-8300 systems
- DMB32-LN for the VAX-8800 system with an expander cabinet

The only difference between the three versions is the length of the six ribbon cables that go between the VAXBI backplane and the H3033 distribution panel. The contents of the three kits are given in Table 2-1.

Table 2-1 DMB32 Installation Kit Details

Part Number	Description	DMB32-LJ	-LM	-LN
T1012	DMB32 module	1	1	1
H3033	Distribution panel	1	1	1
12-22246-01	Transition header assembly	1	1	1
17-00740-02	5 ft ribbon cable assembly	6	0	0
17-00740-04	8 ft ribbon cable assembly	0	6	0
17-00740-05	15 ft ribbon cable assembly	0	0	6
12-24866-01	H3195 unbalanced sync loopback	1	1	1
12-24867-01	H3196 balanced sync loopback	1	1	1
12-15336-07	H3197 async loopback	1	1	1
EK-DMB32-UG	DMB32 User Guide	1	1	1

2.5 INSTALLATION CHECKS

When the DMB32 hardware has been installed, the DMB32 synchronous device driver must be installed before the sync port can be used. The DMB32 synchronous driver is a layered product that must be ordered separately from DIGITAL, it is not part of the DMB32 hardware kits described in section 2.4.

The procedure for installing the DMB32 Sync Driver is fully described in the DMB32 Sync Driver Installation Guide supplied in the software installation kit.

NOTE

VAX/VMS requires that an adapter cable is connected to the sync port before it will configure the sync device (SIx0).

2.6 CONFIGURATION RULES

In VAXBI systems the base address of each node is defined by the node ID plug fitted to the backplane. At the base address, the node must identify its device type and its revision level. Therefore, there is no need for complex fixed, or floating, address schemes, such as are required on UNIBUS and Q-bus systems.

There are, however, two VAXBI configuration rules which do apply to the DMB32.

1. The DMB32 must have a unique node ID (set by the node ID plug). The DWBUA UNIBUS adapter is always assigned a node ID of zero; therefore, if the host system has a DWBUA installed, the DMB32 cannot be node zero.

2. The DMB32 must not be installed in slot 1 of the first backplane. This is also called slot K1J1 and is reserved for the node which supplies the BI TIME and BI PHASE clock signals.

There may also be system-dependent configuration restrictions, such as those imposed by backplane capacity, and physical mounting limitations within the cabinet. These limitations will be described in the host system documentation.

2.7 MECHANICAL INSTALLATION

Figure 2-1 shows how the parts of the DMB32 option fit together. This figure should be used together with the installation instructions given in this section.

WARNING

Shut off the system power and disconnect the main system power cord before performing any procedure in this chapter.

2.7.1 Electro-Static Discharge Precautions

CAUTION

You must wear an anti-static wrist strap connected to an active ground whenever you work on a system with the covers removed, or handle the T1012 module.

2.7.1.1 Anti-Static Wrist Strap – The anti-static wrist strap is located within the system cabinet of the host system (refer to the host system documentation for a description of where to find the wrist strap), and is connected to ground. Place this wrist strap on your wrist before performing any of the following procedures.

2.7.1.2 Conductive Module Containers – Do not remove the T1012 module from its conductive container until you are ready to install it into the cardcage. Whenever you remove a VAXBI module from the cardcage, place it in a conductive container.

2.7.2 T1012 Module Installation

1. Insert the T1012 into a slot in the VAXBI cardcage.

   The module may be installed in any empty slot except slot K1J1 (that is, the first slot in the first backplane). The module is keyed to prevent incorrect installation.

2. Fit an appropriate node ID plug to the reverse side of the backplane at the position corresponding with the slot where the T1012 module is inserted.

   Take care not to bend the pins of the node ID plug.

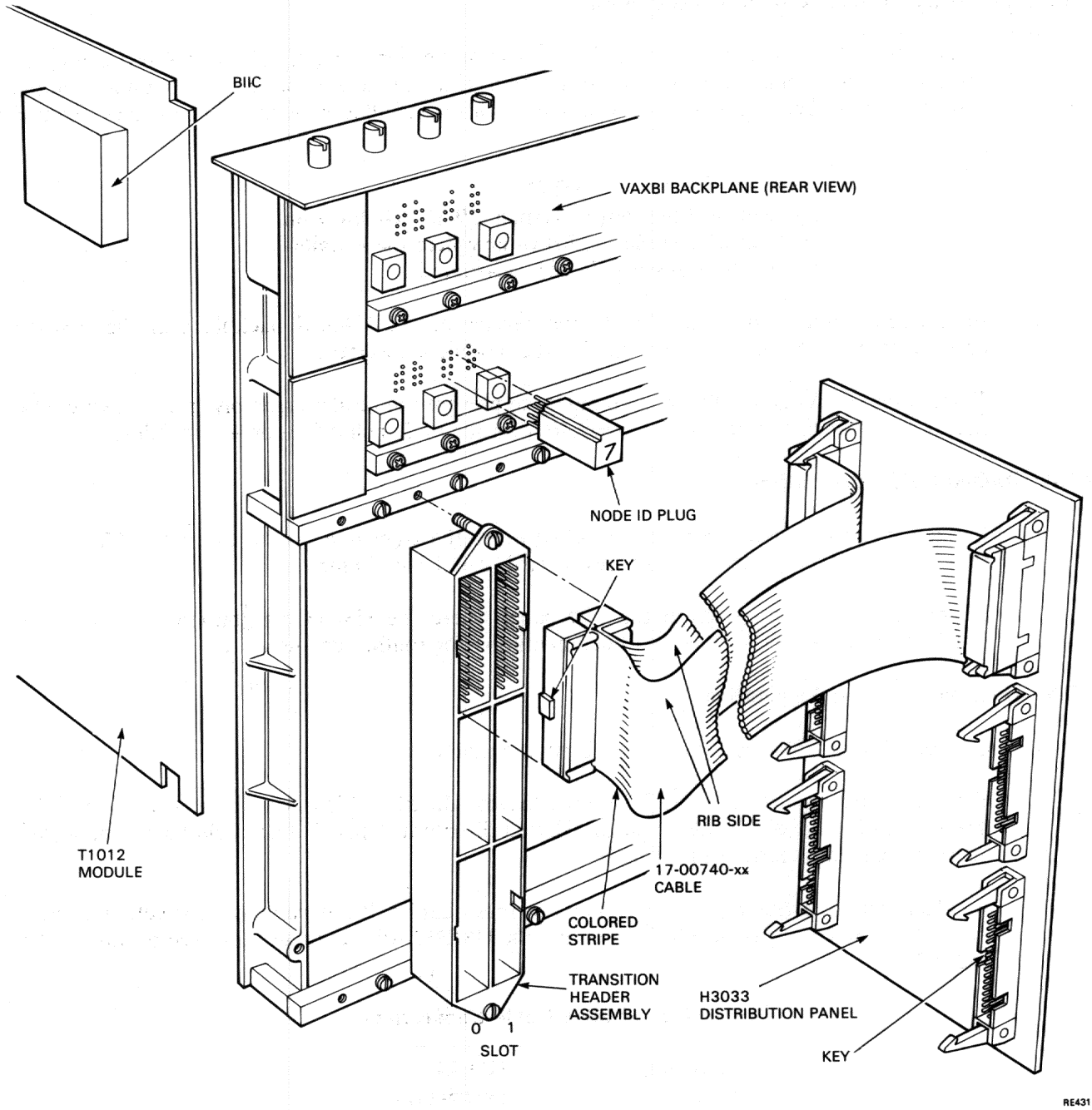


Figure 2-1 DMB32 Installation

2.7.3 Transition Header Assembly Installation

1. Check the VAXBI backplane to see if a transition header assembly (12-22246-01) is already installed on the backplane at the correct position. If it is, you will not need to continue with the procedure described in this section. Continue with the installation of the ribbon cables (Section 2.7.4).

NOTE
You will need the torque wrench (29-17381-00) from the VAXBI installation kit to complete the installation of the transition header assembly.

2. Fit the transition header assembly to the reverse of the VAXBI backplane at the position corresponding with the slot where the T1012 module is inserted.

3. Tighten the screws gradually and alternately to prevent the header assembly from skewing. Use the torque wrench to tighten the screws to a torque of 5 inch.lbf (+ or – 1 inch.lbf).

2.7.4 Ribbon Cable Installation

1. Connect the six ribbon cables (17-00740-xx) into the transition header assembly. Make sure that the ribbed side of the cables faces towards slot 1 (see Figure 2-1).

2. Route the six ribbon cables to the I/O panel where the H3033 distribution panel is to be installed. Refer to the host system documentation for routing details.

2.7.5 Distribution Panel Installation

1. Connect the six ribbon cables to the distribution panel.

   Table 2-2 and Figures 2-2 and 2-3 will help you identify which connector on the transition header connects to which socket on the H3033 distribution panel. The cable header plugs are keyed to prevent incorrect installation.

   An electrical key signal passes through all six ribbon cables. If any cable is incorrectly installed, the path of the key signal will be broken and the DMB32 will fail its self-test (see Sections 2.8.1 and 4.3.2).

Table 2-2 Ribbon Cable Connections

Transition Header Connector	H3033 Distribution Panel Socket
C-1	J10
C-2	J13
D-1	J11
D-2	J14
E-1	J12
E-2	J15

2. Install the H3033 distribution panel into the appropriate position on the system cabinet.

3. Fit the adapter cable, line printer cable, and async cables (as appropriate) to the distribution panel. Adapter cables are fully described in Section 2.9, Cabling.

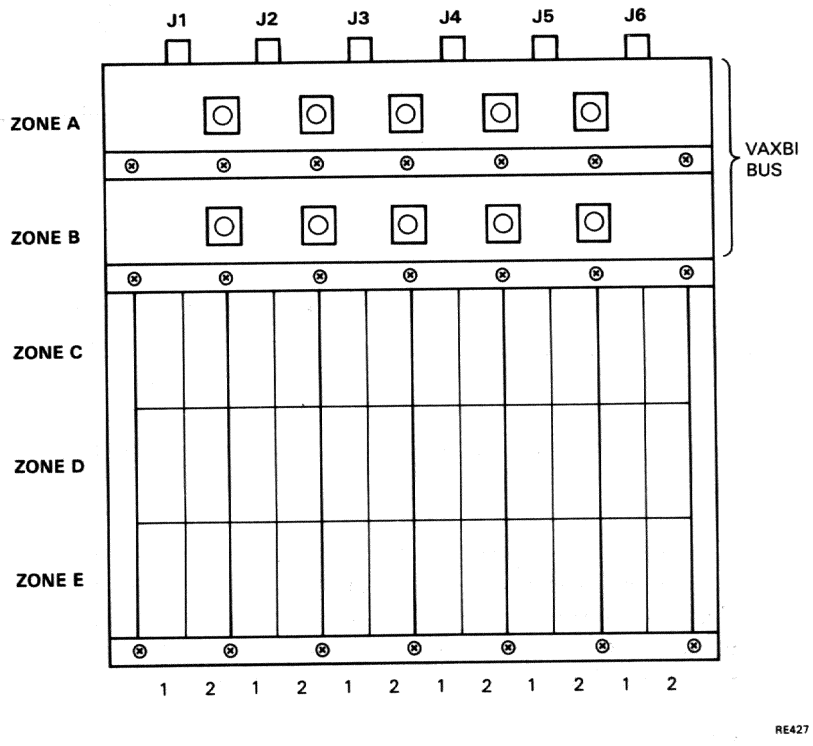


Figure 2-2 The VAXBI Backplane (Rear View)

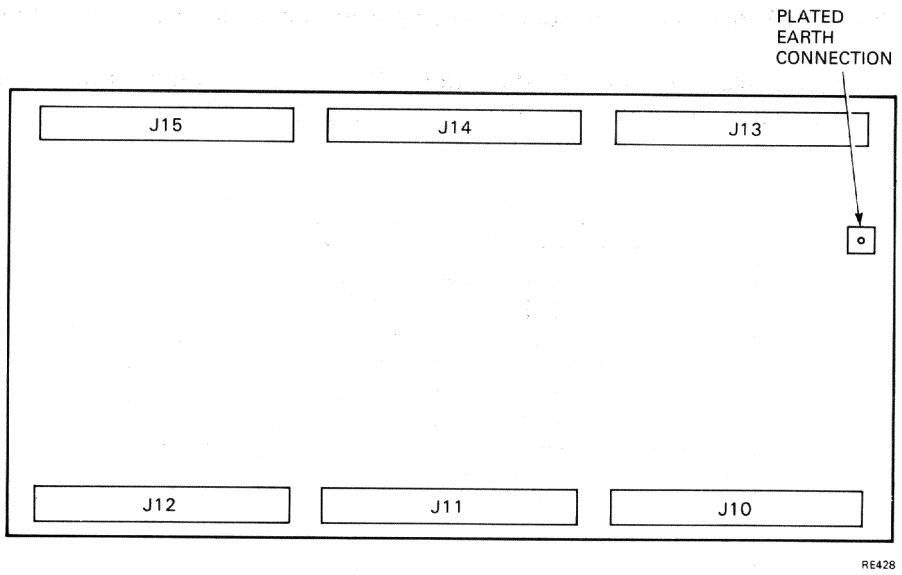


Figure 2-3 The H3033 Distribution Panel (Rear View)

NOTE

VAX/VMS requires that an adapter cable is connected to the sync port before it will configure the sync device (SIx0). It also requires that a line printer cable is connected to the printer port before it will configure the printer device (LIx0).

2.8 ACCEPTANCE TESTING

When you have installed the DMB32, run the following tests.

1. Self-test (runs on power-up)
2. EVDAK, EVDAJ, EVDAL, EVAAA diagnostics
3. UETP

The sequence used to test the DMB32 is described in the following sections.

2.8.1 Power-Up and Self-Test

1. Power-up the host system.

2. Check that the yellow LED on the T1012 module lights after about four seconds, and stays lit. This indicates a successful self-test.

3. On the VAX-8200 and VAX-8300, check the printout from the system console. Immediately after its own CPU self-test, the host system prints a sequence of numbers and periods, for example:

```
0 . 2 . . . . 7 . . . B C D . .
```

   This example indicates that the host has detected modules with node IDs of 0, 2, 7, 11, 12, and 13, and that they have passed their self-test diagnostics. If a module fails its self-test, it is not configured by the host and the host puts a minus sign in front of the node ID, for example:

```
0 . 2 . . . . 7 . . .-B C D . .
```

   This example indicates that the option with a node ID of B_{16} has not passed its self-test.

   The host should recognize that a module is present at the node ID with which you have installed the T1012 module.

4. If the DMB32 self-test fails, type the following at the console prompt:

```
>>> E/P/W 200xx000 <RET>
```

   (where xx is twice the node ID)

   The console should respond with 0109_{16}, the valid DMB32 device type.

   If the console reports an error, check that you have the entered the correct address for the node ID.

   If the console responds with $FFFF_{16}$, try reseating the T1012 module.

   If the console gave the correct device type (0109_{16}), type the following at the console prompt:

```
>>>> E/P/L 200xx0F0
```

   (where xx is twice the node ID)

   The console will print out the contents of the GPR0 register in the DMB32 which contains bits that indicate the reason for the self-test failing (see Section 3.3.9 for an interpretation of these bits).

2.8.2 Diagnostics

After a successful self-test, use the following diagnostic sequence to make sure that the DMB32 is fully functional. Full details of these diagnostics and how to run them are given in Chapter 4, Maintenance. Run each diagnostic for at least one error-free pass.

1. Load the VAX Diagnostic Supervisor (VDS).
2. Load and run EVDAK, the DMB32 level 3 diagnostic.
3. Boot the VAX/VMS system.
4. Load and run EVDAJ, the DMB32 level 2R async diagnostic.
5. Load and run EVDAL, the DMB32 level 2R sync diagnostic.
6. Load and run EVAAA, the DMB32 level 2R printer diagnostic.
7. Run UETP.

2.9 CABLING

This section contains details of the distribution panel and adapter cables referred to in Section 2.7. These adapter cables are not supplied as part of the DMB32 option, but must be ordered separately. Section 2.7 also describes the loopback connectors, both those supplied with the DMB32 option, and those which are not supplied but are needed to test the adapter cables.

2.9.1 H3033 Distribution Panel

Figure 2-4 shows a front view of the H3033 distribution panel.

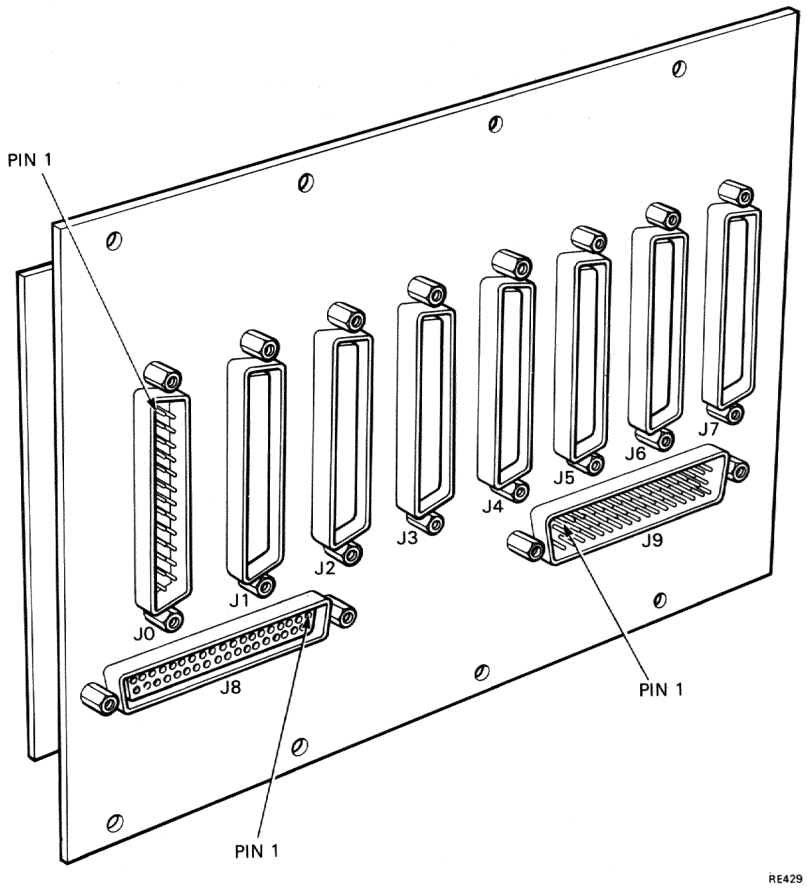


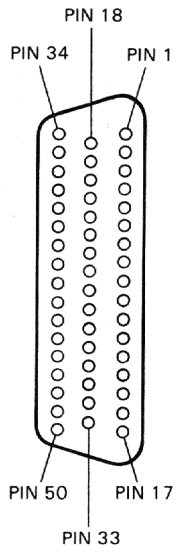
Figure 2-4 H3033 Distribution Panel (Front View)

2.9.2 Adapter Cables

Adapter cables for V.35, V.24, RS-422, and RS-423-A are available for use on the 50-way sync connector of the distribution panel. Pinout information of the sync connector is repeated here for reference. Interconnection and pinout detail for each adapter cable follows.

50-WAY PIN	SIGNAL NAME
1	CODE GROUND
2	CODE 0
3	CODE 1
4	CODE 2
5	CODE 3
6	Tx DATA (A)
7	Tx DATA (B)
8	Tx DATA
9	RTS/C (A)
10	RTS/C (B)
11	Rx DATA (A)
12	Rx DATA (B)
13	LOCAL LOOP
14	SPEED INDICATE
15	TEST I
16	REM.LOOP
17	RI
18	Rx CLOCK (A)
19	Rx CLOCK (B)
20	Tx CLOCK (A)
21	Tx CLOCK (B)
22	CLOCK
23	V.35 Tx CLOCK (A)
24	V.35 Tx CLOCK (B)
25	V.35 CLOCK (A)
26	V.35 CLOCK (B)
27	V.35 Rx DATA (A)
28	V.35 Rx DATA (B)
29	V.35 Tx DATA (A)
30	V.35 Tx DATA (B)
31	V.35 Rx CLOCK (A)
32	V.35 Rx CLOCK (B)
33	DTR
34	DSR (A)
35	DSR (B)
36	RTS
37	DCD/I (A)
38	DCD/I (B)
39	CTS (A)
40	CTS (B)
41	DCE GROUND
42	TEST 1
43	TEST 2
44	DTE GROUND
45	DTR (A)
46	DTR (B)
47	CLOCK (A)
48	CLOCK (B)
49	TEST 3
50	SPEED

(A),(B) WIRES A(+ve) AND B(- ve) OF A TWISTED PAIR



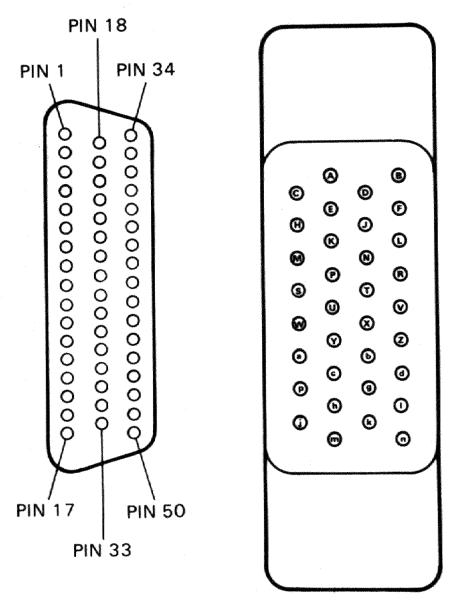
50-WAY D-TYPE CONNECTOR (MALE PLUG)

RE146

Figure 2-5 50-Way Sync Channel Connector

2.9.2.1 V.35 Adapter Cable

50-WAY PINS	SIGNAL NAME	34-WAY PINS
1	CODE GROUND	*
2	CODE 0	*
3	CODE 1	
4	CODE 2	
5	CODE 3	
17	RI	J
23	V.35 Tx CLOCK (A)	Y
24	V.35 Tx CLOCK (B)	a
25	V.35 CLOCK (A)	U
26	V35 CLOCK (B)	W
27	V.35 Rx DATA (A)	R
28	V.35 Rx DATA (B)	T
29	V.35 Tx DATA (A)	P
30	V.35 Tx DATA (B)	S
31	V.35 Rx CLOCK (A)	V
32	V.35 Rx CLOCK (B)	X
33	DTR	H
34	DSR (A)	E
35	DSR (B)	#
36	RTS	C
37	DCD/I (A)	F
38	DCD/I (B)	#
39	CTS (A)	D
40	CTS (B)	#
41	DCE GROUND	#
44	DTE GROUND	A,#



50-WAY D-TYPE CONNECTOR (FEMALE) **34-WAY SQUARE CONNECTOR (MALE)**

* CONNECTED TOGETHER

\# CONNECTED TO DCE GROUND

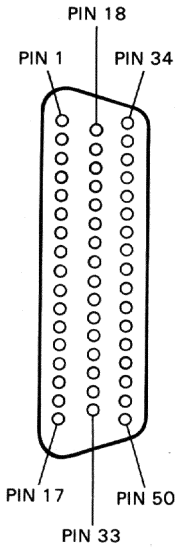
(A),(B) WIRES A(+ve) AND B(-ve) OF A TWISTED PAIR

RE147

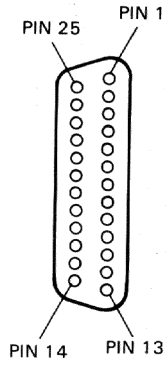
Figure 2-6 BC19F-02 (17-01112-01) Adapter Cable Detail

2.9.2.2 V.24 Adapter Cable

50-WAY PINS	SIGNAL NAME	25-WAY PINS
1	CODE GROUND	*
2	CODE 0	
3	CODE 1	*
4	CODE 2	
5	CODE 3	
8	Tx DATA	2
11	Rx DATA (A)	3
12	Rx DATA (B)	#
13	LOCAL LOOP	18
15	TEST I	25
16	REM.LOOP	21
17	RI	22
18	Rx CLOCK (A)	17
19	Rx CLOCK (B)	#
20	Tx CLOCK (A)	15
21	Tx CLOCK (B)	#
22	CLOCK	24
33	DTR	20
34	DSR (A)	6
35	DSR (B)	#
36	RTS	4
37	DCD/I (A)	8
38	DCD/I (B)	#
39	CTS (A)	5
40	CTS (B)	#
41	DCE GROUND	#
44	DTE GROUND	7,#
50	SPEED	23



50-WAY D-TYPE CONNECTOR (FEMALE)



25-WAY D-TYPE CONNECTOR (MALE)

* – CONNECTED TOGETHER

– CONNECTED TO DCE GROUND

(A),(B) – WIRES A(+ve) AND B(– ve) OF A TWISTED PAIR

RE149

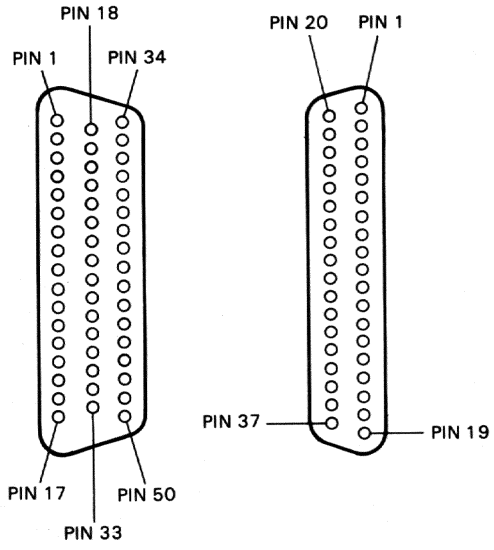
Figure 2-7 BC19D-02 (17-01110-01)Adapter Cable Detail

2.9.2.3 RS-422 Adapter Cable

50-WAY PINS	SIGNAL NAME	37-WAY PINS
1	CODE GROUND	*
2	CODE 0	
3	CODE 1	
4	CODE 2	*
5	CODE 3	
6	Tx DATA (A)	4
7	Tx DATA (B)	22
9	RTS/C (A)	7
10	RTS/C (B)	25
11	Rx DATA (A)	6
12	Rx DATA (B)	24
13	LOCAL LOOP	10
14	SPEED INDICATE	2
15	TEST I	18
16	REM.LOOP	14
17	RI	15
18	Rx CLOCK (A)	8
19	Rx CLOCK (B)	26
20	Tx CLOCK (A)	5
21	Tx CLOCK (B)	23
34	DSR (A)	11
35	DSR (B)	29
37	DCD/I (A)	13
38	DCD/I (B)	31
39	CTS (A)	9
40	CTS (B)	27
41	DCE GROUND	20
44	DTE GROUND	19, 37
45	DTR (A)	12
46	DTR (B)	30
47	CLOCK (A)	17
48	CLOCK (B)	35
50	SPEED	16

* – CONNECTED TOGETHER

(A),(B) – WIRES A(+ve) AND B(-ve) OF A TWISTED PAIR



50-WAY D-TYPE CONNECTOR (FEMALE)

37-WAY D-TYPE CONNECTOR (MALE)

RE150

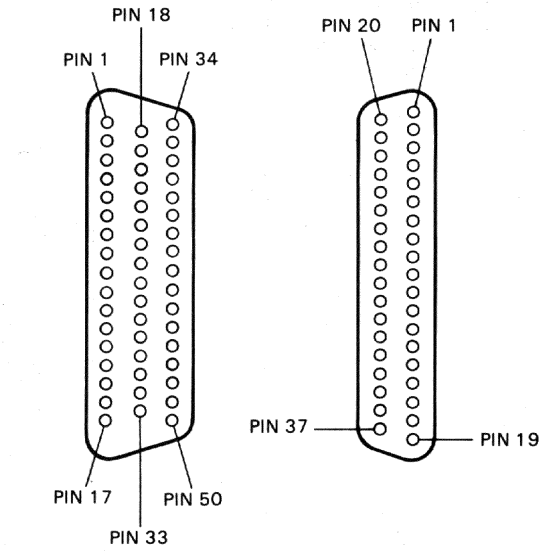
Figure 2-8 BC19B-02 (17-01108-01) Adapter Cable Detail

2.9.2.4 RS-423-A Adapter Cable

50-WAY PINS	SIGNAL NAME	37-WAY PINS
1	CODE GROUND	*
2	CODE 0	
3	CODE 1	*
4	CODE 2	
5	CODE 3	
8	Tx DATA	4
11	Rx DATA (A)	6
12	Rx DATA (B)	24
13	LOCAL LOOP	10
14	SPEED INDICATE	2
15	TEST I	18
16	REM.LOOP	14
17	RI	15
18	Rx CLOCK (A)	8
19	Rx CLOCK (B)	26
20	Tx CLOCK (A)	5
21	Tx CLOCK (B)	23
22	CLOCK	17
33	DTR	12
34	DSR (A)	11
35	DSR (B)	29
36	RTS	7
37	DCD/I (A)	23
38	DCD/I (B)	32
39	CTS (A)	9
40	CTS (B)	27
41	DCE GROUND	20
44	DTE GROUND	19, 22, 25, 30, 35, 37
50	SPEED	16

* – CONNECTED TOGETHER

(A), (B) – WIRES A(+ve) AND B(–ve) OF A TWISTED PAIR



RE151

Figure 2-9 BC19E-02 (17-01111-01) Adapter Cable Detail

2.9.3 Loopback Connectors

– A range of loopback connectors is available for testing the DMB32. Three are supplied with the option: H3197 is used to test a single async channel, H3195 and H3196 are used to test the sync channel. The other loopback connectors are used to test the adapter cables and modem cables, and are not supplied with the DMB32 option. The full list of loopback connectors is:

- H3196 50-way balanced (sync channel)
- H3195 50-way unbalanced (sync channel)
- H3197 25-way (async channel)

} Supplied with the option

- H3250 34-way V.35 (sync channel)
- H3248 25-way V.24 (sync channel)
- H3198 37-way RS-422 and RS-423-A (sync channel)

2.9.3.1 H3196 50-Way Balanced Loopback Connector – This loopback connector is used to test all the balanced, and many of the unbalanced, drivers and receivers. (DTR, CTS, RTS, and DCD are tested with both the unbalanced and balanced loopbacks.)

The balanced loopback is a 50-way D-type connector that plugs into the sync port on the distribution panel, and is wired as given in Table 2-3.

Table 2-3 H3196 Balanced Loopback Interconnections

Pins Connected Together	Signals
1,3,4,5	Connector identity code
35,41,44	Ground and receiver inputs
6,11	Data A
7,12	Data B
9,37	RTS/C A, DCD/I A
10,38	RTS/C B, DCD/I B
13,15	Local loop, Test indicator
50,17,14	Speed, Ring indicator, and Speed indicator
16,34	Remote loop, DSR A
47,18,20	Clock A
48,19,21	Clock B
45,39	DTR A, CTS A
46,40	DTR B, CTS B
29,27	V.35 data A
30,28	V.35 data B
25,23,31	V.35 clock A
26,24,32	V.35 clock B

2.9.3.2 H3195 50-Way Unbalanced Loopback Connector – This loopback connector is used to test all the unbalanced drivers and receivers. It is a 50-way D-type connector that plugs into the sync port on the distribution panel, and is wired as given in Table 2-4.

Table 2-4 H3195 Unbalanced Loopback Interconnections

Pins Connected Together	Signals
1,2,5	Connector identity code
12,19,21,35,38,40,41,44	Ground and receiver B inputs
8,11	Data
13,15	Local loopback, Test indicator
16,34	Remote loop, DSR
50,17,14	Speed, Ring indicator, and Speed indicator
22,18,20	Clock
33,39	DTR, CTS
36,37	RTS, DCD

2.9.3.3 H3197 25-Way Loopback Connector (Async Channels) – This connector is used to test a single asynchronous channel. (Eight H3197s will be needed to test all the async channels simultaneously.) It is a 25-way D-type connector that plugs into an async port on the distribution panel, and is wired as shown in Figure 2-10.

The wiring of the H3197 loopback is similar to that in the H325 async loopback (used with most other async communications options), and the H325 connector can be used to test the DMB32. However, because of the physical layout of the distribution panel, it is not possible to connect H325 loopbacks to all eight ports at the same time, nor to connect them to J5, J6, or J7 without first removing the sync port cable.

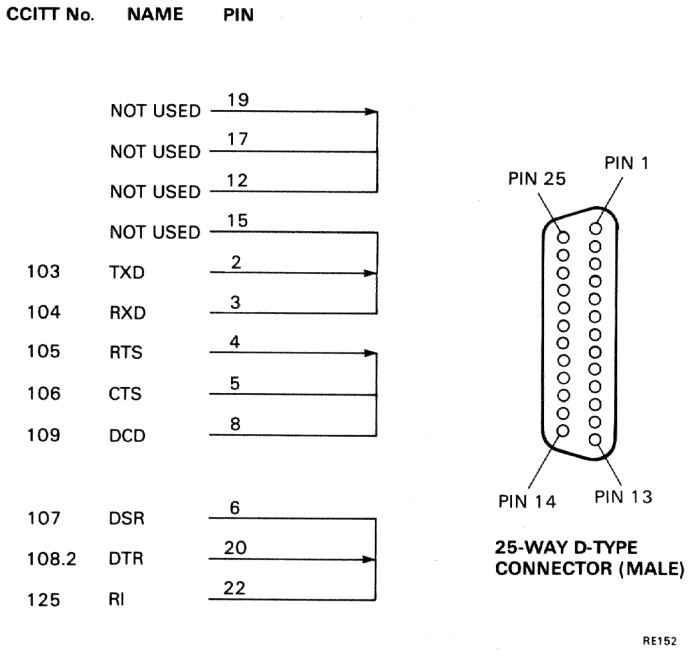


Figure 2-10 H3197 Loopback Connector

2.9.3.4 H3250 34-Way Loopback Connector (V.35) – This connector is used to test the V.35 interface of the sync channel and the V.35 adapter cable. It is a 34-way square connector that attaches to the end of the V.35 adapter cable, and is wired as shown in Figure 2-11.

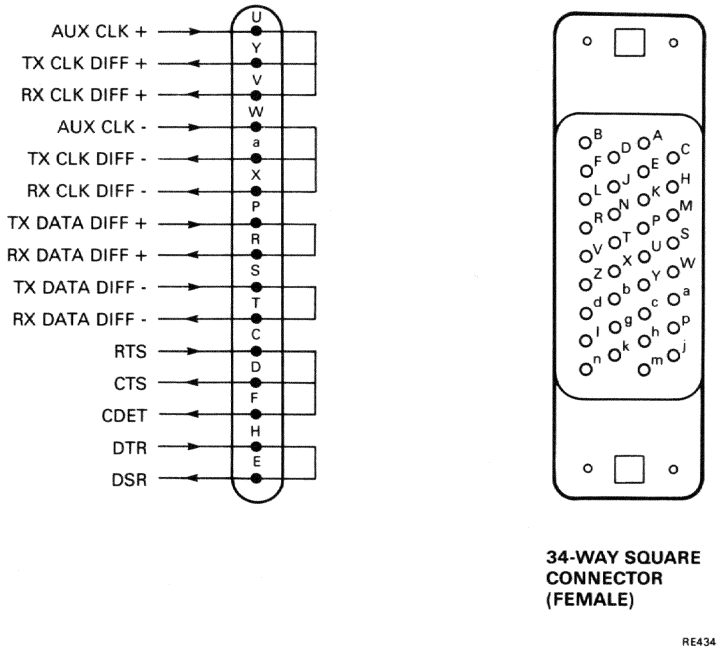


Figure 2-11 H3250 Loopback Connector

2.9.3.5 H3248 37-Way Loopback Connector (V.24) – This connector is used to test the V.24 interface of the sync channel and the V.24 adapter cable. It is a 37-way D-type connector, and is wired as shown in Figure 2-12.

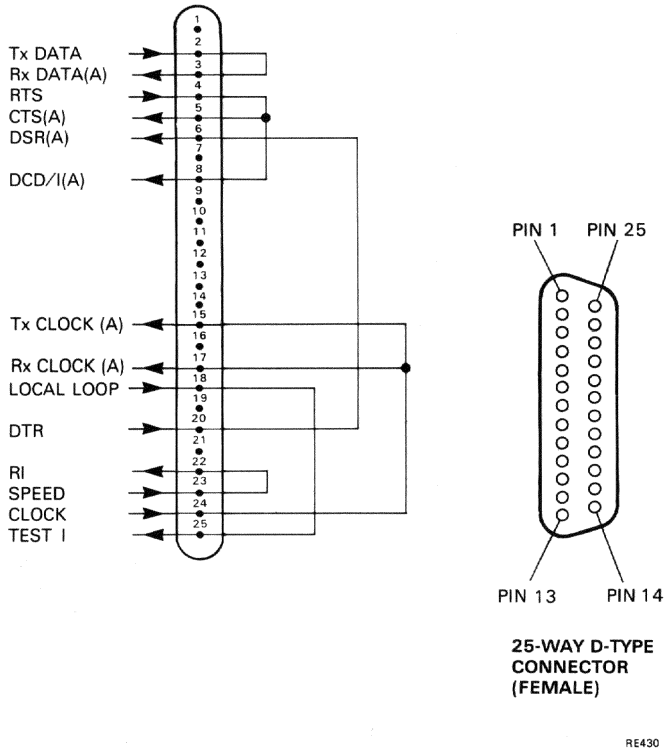


Figure 2-12 H3248 Loopback Connector

2.9.3.6 H3198 37-Way Loopback Connector (RS-422/423) – This connector is used to test the RS-422 and RS-423 interfaces of the sync channel and the RS-422 and RS-423 adapter cables. It is a 37-way D-type connector, and is wired as shown in Figure 2-13.

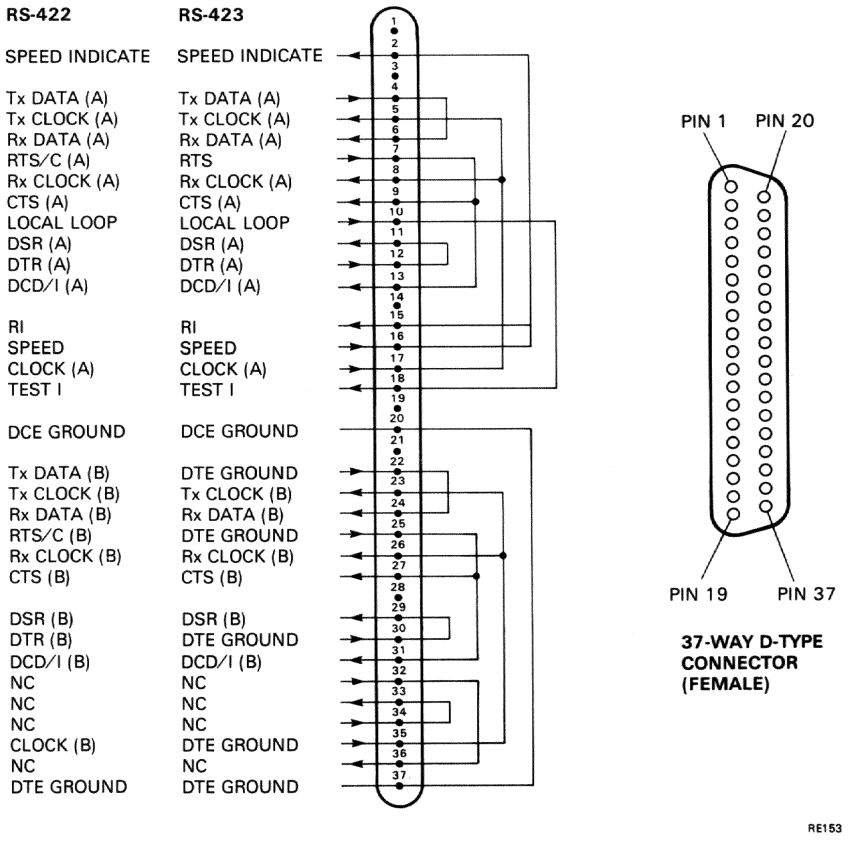


Figure 2-13 H3198 Loopback Connector

CHAPTER 3
OPERATION AND PROGRAMMING

3.1 SCOPE

This chapter describes the device registers, and how they are used to control and monitor the DMB32. The chapter covers:

- The register map
- The bit functions and format of each register
- Programming features available to the host

3.2 OPERATION

The host system controls and monitors the DMB32 option by using registers in the BIIC and CASRAM. Command longwords, words, or bytes are written to the registers by the VAXBI host, and are interpreted and executed by the DMB32 firmware. Status reports and data to the host are also transferred using the registers.

3.2.1 DMB32 Register Map

The DMB32 occupies 8 Kbytes of VAXBI memory-mapped I/O space. During normal operation, all commands to the DMB32 are made through the device registers in this address space.

The 'Base' address of the 8 Kbytes within the VAXBI bus I/O space is determined by the 'Node ID' plug which is fitted to the VAXBI backplane. There are no option switches on the module itself.

The set of DMB32 device registers is composed of two major groups. The first group includes the VAXBI registers and the second includes all the DMB32-specific registers. These can be further subdivided by function, as shown in Figure 3-1.

Table 3-1 lists each DMB32 device register and gives its type and address offset from the 'Base' address of the node. This address is the lowest I/O address to which the option responds (address bits <11:0> are zero), and is determined by the Node ID plug.

The designation 'I' (for 'Indexed') in the 'Address Offset' column indicates that a set of identical indexed registers (one for each channel) are accessed at that location (see Section 3.2.2).

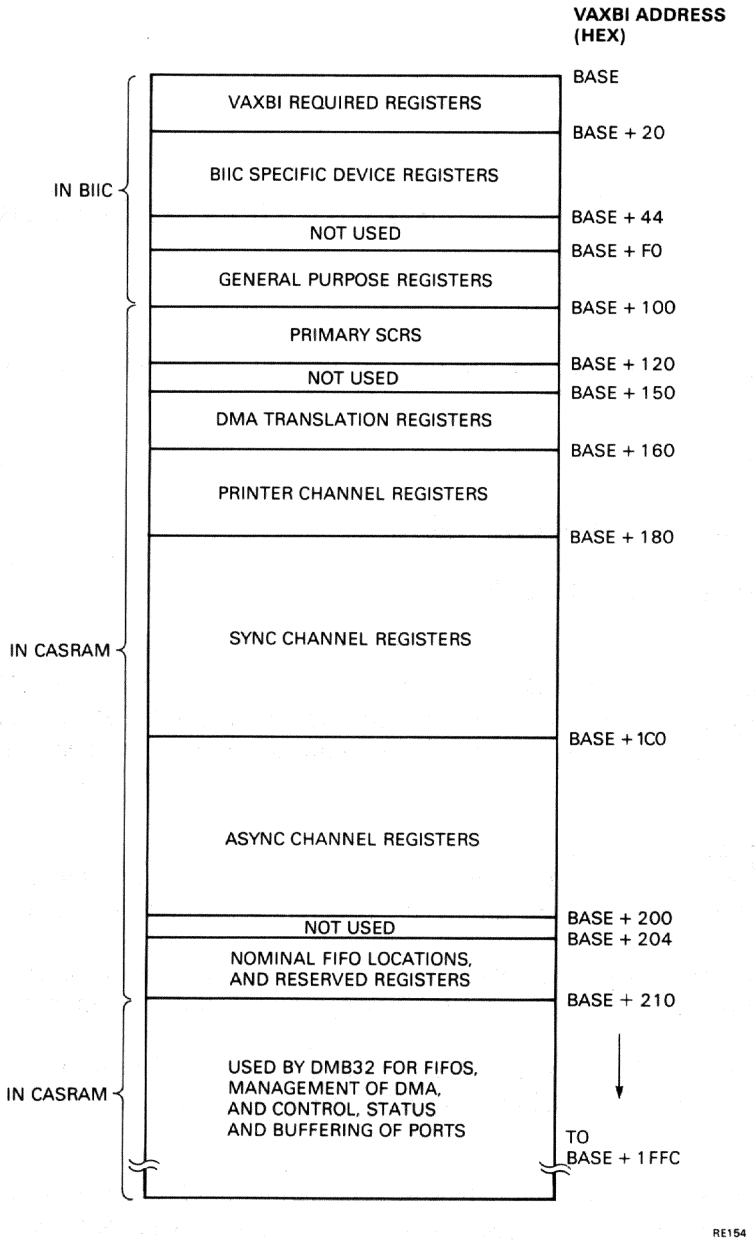


Figure 3-1 DMB32 Register Map

The register type is given as R/W (Read/Write), R (Read Only) or NU (Not Used).

Table 3-1 DMB32 Registers

Register Description	Name	Address Offset (Hexadecimal)	Type
Device type register	DTYPE	0	R/W
VAXBI control and status register	VAXBICSR	4	R/W
Bus error register	BER	8	R
Error interrupt control register	EINTRCSR	C	R/W
Interrupt destination register	INTRDES	10	R/W
IPINTR mask register	IPINTRMSK	14	R/W
Force-Bit IPINTR/STOP destination	FIPSDES	18	R/W
IPINTR source register	IPINTRSRC	1C	R/W
Starting address register	SADR	20	R/W
Ending address register	EADR	24	R/W
BCI control register	BCICSR	28	NU
Write status register	WSTAT	2C	NU
Force-Bit IPINTR/STOP command register	FIPSCMD	30	NU
User interface interrupt control	UINTRCSR	40	R/W
General-purpose register 0	GPR0	F0	R/W
General-purpose register 1	GPR1	F4	NU
General-purpose register 2	GPR2	F8	NU
General-purpose register 3	GPR3	FC	NU
Maintenance register	MAINT	100	R/W
Async control and status	ACSR	104	R/W
Sync control and status	ACSR	108	R/W
Printer control and status	PCSR	10C	R/W
Device configuration	CONFIG	114	R
2nd async control and status	ACSR2	118	R/W
2nd sync control and status	SCSR2	11C	R/W
2nd printer control and status	PCSR2	120	R/W
System page table register	SPTE	150	R/W
System page table size	SPTS	154	R/W
Global page table register	GPTE	158	R/W
Global page table size	GPTS	15C	R/W
Printer prefix/suffix control	PFIX	160	R/W
Printer buffer address	PBUFFAD	164	R/W
Printer buffer count	PBUFFCT	168	R/W
Printer control	PCTRL	16C	R/W
Printer carriage counter	PCAR	170	R/W
Printer page size descriptor	PSIZE	174	R/W
Transmit buffer address	TBUFFAD1	180 (I)	R/W
Transmit buffer count/offset 1	TBUFFCT1	184 (I)	R/W
Receive buffer address 1	RBUFFAD1	188 (I)	R/W
Receive buffer count/offset 1	RBUFFCT1	18C (I)	R/W
Buffer 1 transmit control	TLNCTRL1	190 (I)	R/W
Buffer 1 receive control	RLNCTRL1	194 (I)	R/W
Sync line parameters 1	LPR1	198 (I)	R/W
Sync line parameters 2	LPR2	19C (I)	R/W
Transmit buffer address 2	TBUFFAD2	1A0 (I)	R/W

Table 3-1 DMB32 Registers (continued)

Register Description	Name	Address Offset (Hexadecimal)	Type
Transmit buffer count/offset 2	TBUFFCT2	1A4 (I)	R/W
Receive buffer address 2	RBUFFAD2	1A8 (I)	R/W
Receive buffer count/offset 2	RBUFFCT2	1AC (I)	R/W
Buffer 2 transmit control	TLNCTRL2	1B0 (I)	R/W
Buffer 2 receive control	RLNCTRL2	1B4 (I)	R/W
Sync line parameters 3	LPR3	1B8 (I)	R/W
Sync buffer control bits	BUFCTRL	1BC (I)	R/W
Transmission preempt buffer	PREEMPT	1C0 (I)	R/W
Transmit buffer address	TBUFFADD	1C4 (I)	R/W
Transmit buffer count/offset	TBUFFCT	1C8 (I)	R/W
Line parameter register	LPR	1CC (I)	R/W
Line control register	LNCTRL	1D0 (I)	R/W
Line status register	LSTAT	1D4 (I)	R/W
Flow control characters	FLOWC	1D8 (I)	R/W
Reserved		1DC to 1FC	
Receive console data register	–	200	NU
Transmit completion FIFO	TBUF	204	R
Sync line completion FIFO	SBUF	208	R
Receive buffer	RBUF	20C	R

3.2.2 Register Access

1. Registers marked (I) are implemented as a set of indexed registers, one for each channel. When an (I) register is accessed, the address is internally indexed by the ASYNC.IND.ADD parameter in the async CSR (for async registers), or by the SYNC.IND.ADD parameter in the sync CSR (for sync registers), the value of the index parameter being the channel number. Therefore, before (I) registers are accessed, the channel number must be written to the corresponding CSR, as shown in the following example.

   To read the async line parameter register for channel 3, the following commands would be executed:

```
MOVB #CHAN, @#BASE+^X104    ;WRITE CHANNEL NUMBER TO ACSR
MOVL @#BASE+^X1CC, R0        ;READ THE LPR
```

   where CHAN (the channel number) = 3.

2. Not all register bits are specified. In a write action, any unspecified bit must normally be written as a 0. In a read action, unspecified bits are undefined. However, if an unspecified bit is read as a 1, it can be written as a 1 or 0. This allows read-modify-write operations to work correctly.

3. The VAXBI architecture requires that all registers on the VAXBI bus are capable of being read without the contents of the register changing. This means that FIFOs which are 'popped' by a read from the host are not allowed. To overcome this restriction, the registers that are FIFOs are read by the host without removing the FIFO entry (if there is one). To remove the top FIFO entry, the host must write to the location of the FIFO. However, a write must not be done when the FIFO is empty.

4. The register at 'Base + 200' is reserved for the VAXBI console. The DMB32 does not support console functions, therefore this register does not exist (that is, the DMB32 will respond to an access to this address with a NO ACK).

5. Many registers have read-only fields (or fields that may not be written to under some circumstances). The host should make sure that it does not accidentally write to these registers.

6. Register locations between 'Base + 0218' and 'Base + 1FFC' are used as register, FIFO, and buffer space by the DMB32. These locations can only be accessed by the host when the DMB32 is in maintenance mode (MAINT<4> or <5> set).

7. The host should avoid writing to registers unnecessarily. This is because writing to some registers (for example, the parameter registers) causes the on-board microprocessor to examine the registers to determine what action it must take. Such writes, if unnecessary, will degrade the performance of the option.

3.3 REGISTER BIT DEFINITIONS

The following abbreviations are used in the illustration of the registers and the definition of the register bits:

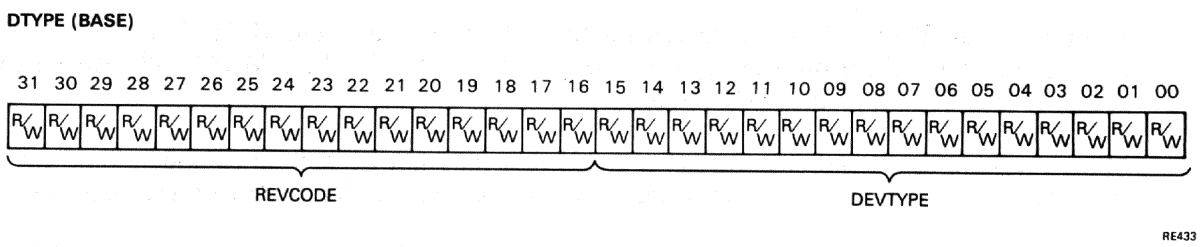
Blank	= Not defined
0	= Read only and always read as 0
1	= Read only and always read as 1
R	= Read only
W	= Write only
R/W	= Read/write
W1C	= Write '1' to clear
*	= Special case, refer to the text
A	= Cleared by Bus Reset only
B	= Cleared by Bus Reset and Programmed Reset
C	= Cleared by Bus Reset, Programmed Reset and Initialization
D	= Cleared by Bus Reset, Programmed Reset, Initialization and Selective Line Reset
Ec	= Cleared
Es	= Set by deassertion of BI_DC_LO.L after a successful self-test
El	= Loaded
F	= Cleared by a STOP command to the DMB32

where:

- Bus Reset includes power-up
- Programmed Reset is MAINT<1>

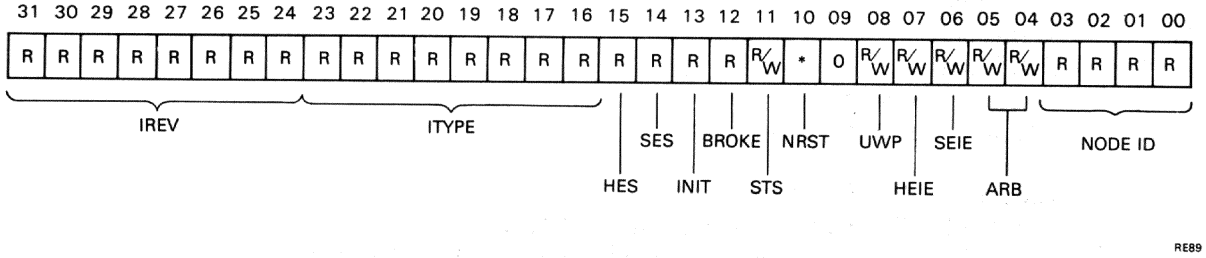
- Initialization is ACSR2<10>, SCSR2<10> and PCSR2<10>
- Selective Line Reset is LPR1<31>

3.3.1 Device Type Register (DTYPE)



Bit	Name	Description
<15:0>	DEVTYPE (Device Type) (R, El)	This field contains 0109_{16} which identifies the device as a DMB32.
<31:16>	REVCODE (Revision Code) (R, El)	This field contains a value that identifies the revision level of the DMB32.

3.3.2 VAXBI Control and Status Register (VAXBICSR)

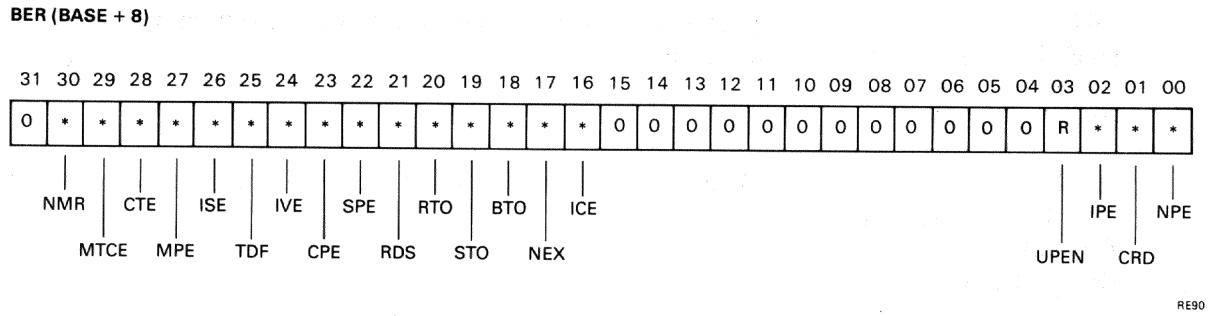


Bit	Name	Description
<3:0>	NODE.ID (Node ID) (R)	This field contains the node ID of the DMB32. It is loaded from the I<3:0> lines at power-up (from the node ID plug).

Bit	Name	Description
<5:4>	ARB (Arbitration Control Bits) (R/W, Ec)	These two bits control the mode of arbitration. 00 – Round robin 01 – High priority 10 – Low priority 11 – No arbitration
<6>	SEIE (Soft Error Interrupt Enable) (R/W, Ec, F)	Determines whether an error interrupt is generated when SES is asserted.
<7>	HEIE (Hard Error Interrupt Enable) (R/W, Ec, F)	Determines whether an error interrupt is generated when HES is asserted.
<8>	UWP (Unlock Write Pending) (W1C, Ec, F)	This bit indicates that a Read Lock transaction has successfully completed, but there has not yet been a Write Unlock command. If a Write Unlock is attempted while this bit is clear, the ISE bit (BUSERR<26>) will be set.
<10>	NRST (Node Reset) (*)	Setting this bit starts the BIIC and DMB32 self-test sequence. This will initialize the DMB32 firmware. This bit will always read as 0.
<11>	STS (Self-Test Status) (R/W, Es)	This bit is cleared when BIIC self-test is started. It is set when the test is passed. If it stays clear, the test has failed.
<12>	BROKE (Broke) (W1C, Ec)	When this bit is set, the DMB32 has not passed the self-test. No other register should be written to when this bit is set. The contents of other registers may not be valid when this bit is set.
<13>	INIT (W1C, Es)	When set, this bit indicates that the DMB32 has not completed its initialization.
<14>	SES (Soft Error Summary) (R)	This bit indicates that one or more soft error bits in the bus error register are set.
<15>	HES (Hard Error Summary) (R)	This bit indicates that one or more hard error bits in the bus error register are set.
<23:16>	ITYPE (Interface Type) (R)	This field indicates the primary interface to the VAXBI. This field reads as 01_{16} in the DMB32.

Bit	Name	Description
<31:24>	IREV (Interface Revision) (R)	This field contains the revision level of the device that provides the primary interface to the VAXBI.

3.3.3 Bus Error Register (BER)

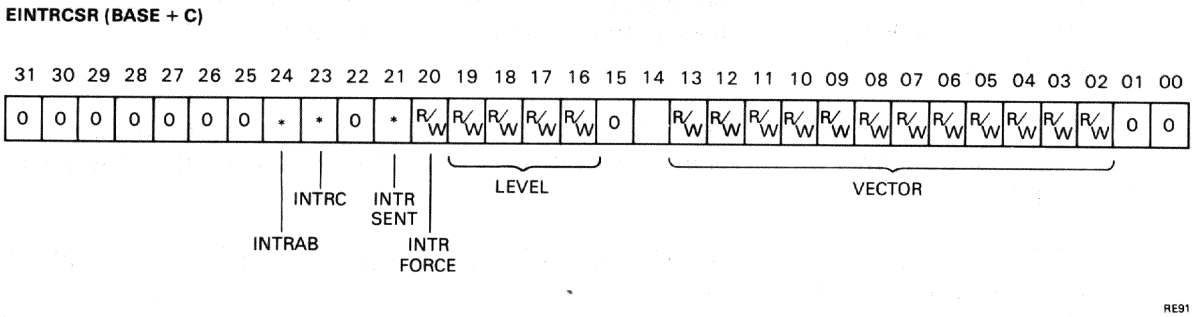


Bit	Name	Description
<0>	NPE (Null Bus Parity Error) (W1C, Ec)	Odd parity was detected on the bus during the second cycle of a two-cycle sequence during which NOARB and BUSY were unasserted.
<1>	CRD (Corrected Read Data) (W1C, Ec)	A CRD status code was received during a READ-type transaction initiated by the master port.
<2>	IPE (ID Parity Error) (W1C, Ec)	A parity error was detected on the encoded master ID during an Imbedded ARB cycle.
<3>	UPEN (User Parity Enable) (R)	Indicates the BIIC parity mode: 1 – User-generated 0 – BIIC-generated It is always zero in the DMB32.
<16>	ICE (Illegal Confirmation Error) (W1C, Ec)	A RESERVED or Illegal Confirmation code was received during a transaction in which the BIIC was involved.
<17>	NEX (Non-Existent Address) (W1C, Ec)	Set when a NO ACK response is received for a READ-type or WRITE-type command sent by the BIIC.
<18>	BTO (Bus Timeout) (W1C, Ec)	Set if the BIIC is unable to start at least one pending transaction before 4096 cycles have elapsed.

Bit	Name	Description
<19>	STO (Stall Timeout) (W1C, Ec)	Set if the slave port asserts the STALL code on the RS<1:0> lines for 128 consecutive cycles.
<20>	RTO (Retry Timeout) (W1C, Ec)	Set if the master receives 128 consecutive RETRY responses from the selected slave for the same master port transaction.
<21>	RDS (Read Data Substitute) (W1C, Ec)	Set if an RDS or RESERVED Status Code is received during a READ-type or IDENT (for vector status) transaction.
<22>	SPE (Slave Parity Error) (W1C, Ec)	Set if the slave port detects a parity error on the bus during a non-ARB cycle of a transaction for which it was selected.
<23>	CPE (Command Parity Error) (W1C, Ec)	Set when the BIIC detects a parity error in a command/address cycle.
<24>	IVE (IDENT Vector Error) (W1C, Ec)	Set by the selected slave if it receives anything but an ACK confirmation from the IDENTing master for the Interrupt Vector.
<25>	TDF (Transmitter During Fault) (W1C, Ec)	Set if the BIIC was driving the VAXBI DAL and I lines (I lines only during the Imbedded ARB cycle) during a cycle that resulted in setting the SPE, MPE, CPE or IPE error bits.
<26>	ISE (Interlock Sequence Error) (W1C, Ec)	Set if the DMB32 successfully completes a Write Unlock transaction when the UWP bit (BICSR<8>) is clear.
<27>	MPE (Master Parity Error) (W1C, Ec)	Set if the master detects a parity error on the bus during a READ-type or vector ACK data cycle.
<28>	CTE (Control Transmitter Error) (W1C, Ec)	Set if the DMB32 detects that its attempt to assert the NO ARB, BSY, or CNF<2:0> lines has failed.
<29>	MTCE (Master Transmit Check Error) (W1C, El)	Set if the master's transmitted data on the DAL, I and P lines does not match the received data.

Bit	Name	Description
<30>	NMR (NO ACK to Multi-Responder Command Received) (W1C, Ec)	Set if the master receives a NO ACK response for an INVAL, STOP, IPINTR, BDCST, or RESERVED command.

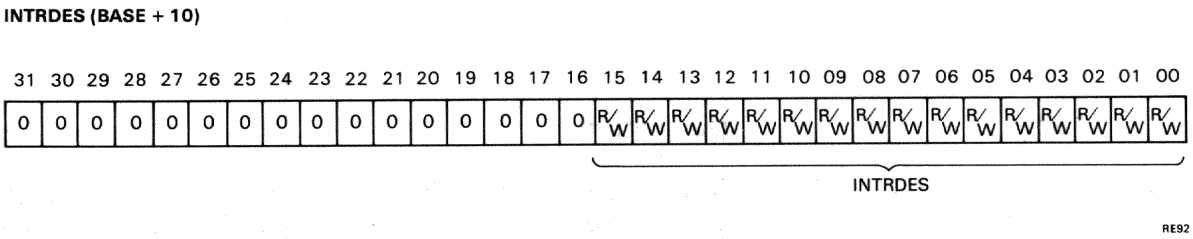
3.3.4 Error Interrupt Control Register (EINTRCSR)



Bit	Name	Description
<13:2>	VECTOR (Error Interrupt Vector) (R/W, Ec)	Vector used during error interrupts. It is transmitted when the DMB32 wins an IDENT ARB cycle on an IDENT that matches the conditions in this control register.
<19:16>	LEVEL (Error Interrupt Level) (R/W, Ec)	Determines the level at which interrupts are transmitted. Also helps to determine whether this register will respond to IDENT commands. If any level bits in the IDENT command match this field (and there is a match in the destination mask), this register will arbitrate for the IDENT.
<20>	INTR.FORCE (Force Error Interrupt) (R/W, Ec, F)	When set, this bit causes an error interrupt request. The DMB32 will set this bit when a DMB32 error has occurred and the error interrupt enable bit is set.
<21>	INTR.SENT (Error Interrupt Sent) (R/W, Ec, F, *)	Indicates that an INTR command has been sent for this interrupt and that an IDENT command is expected. Cleared during an IDENT command after a Level and Master ID match is detected. Also cleared if the error interrupt request is deasserted.
<22>	INTRC (Error Interrupt Complete) (R/W, Ec, *)	Set when the vector for the error interrupt has been transmitted successfully, or has been aborted. Cleared when the error interrupt request is removed.

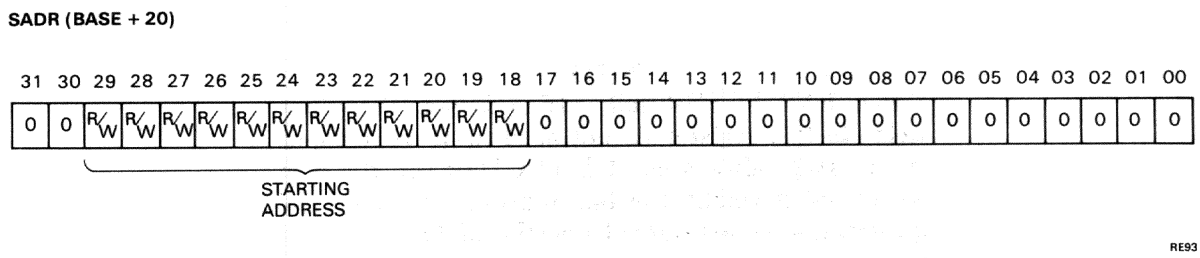
Bit	Name	Description
<23>	INTRAB (Error Interrupt Abort) (R/W, Ec, *)	Set if the error interrupt has aborted. It can only be cleared by the user and has no effect on the BIIC to send further INTR or respond to IDENT commands.

3.3.5 Interrupt Destination Register (INTRDES)



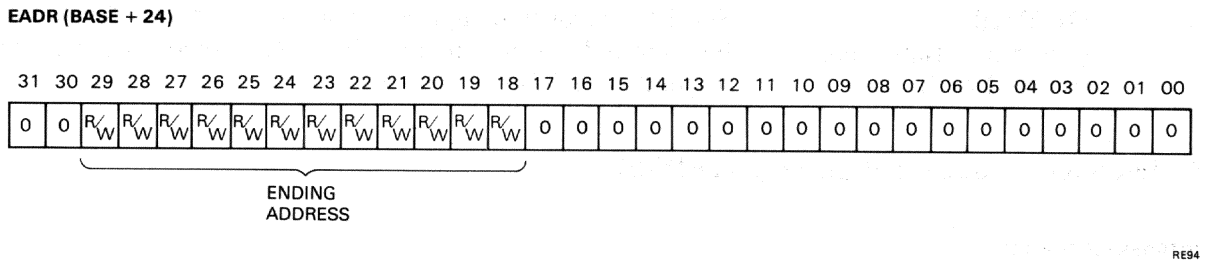
Bit	Name	Description
<15:0>	INTRDES (INTR Destination) (R/W, Ec)	This field is sent out during an INTR command and is used by other nodes to determine whether they should respond. During an IDENT command, the decoded master's ID is compared with this word. If there is a match, the BIIC will respond to the IDENT, provided that there is an interrupt pending that matches the level transmitted in the IDENT command.

3.3.6 Starting Address Register (SADR)



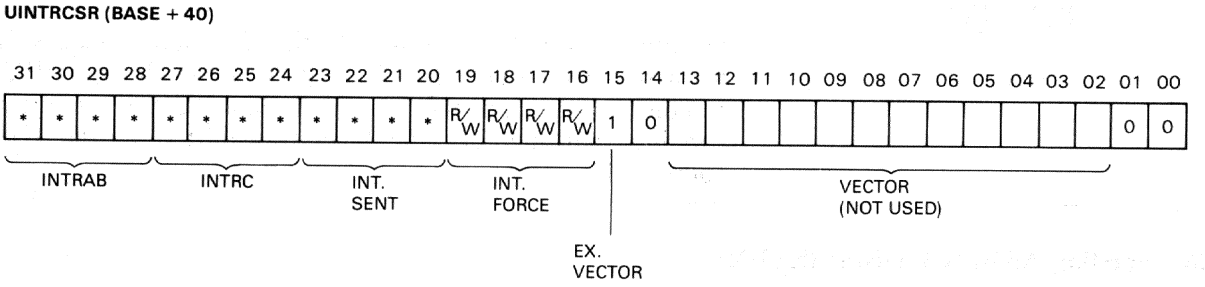
Bit	Name	Description
<29:18>	STARTING. ADDRESS (Starting Address) (R/W, Ec)	Bits <29:18> of the start address of a block of addresses to be recognized by the BIIC as node window space (note that the DMB32 uses window space only for maintenance). The lowest address recognized has bits <17:0> equal to 0.

3.3.7 Ending Address Register (EADR)



Bit	Name	Description
<29:18>	ENDING. ADDRESS (Ending Address) (R/W)	Bits <29:18> of the address of the first location after the block of addresses to be recognized by the BIIC as node window space. The lowest address recognized has bits <17:0> equal to 0.

3.3.8 User Interface Interrupt Control Register (UINTRCSR)



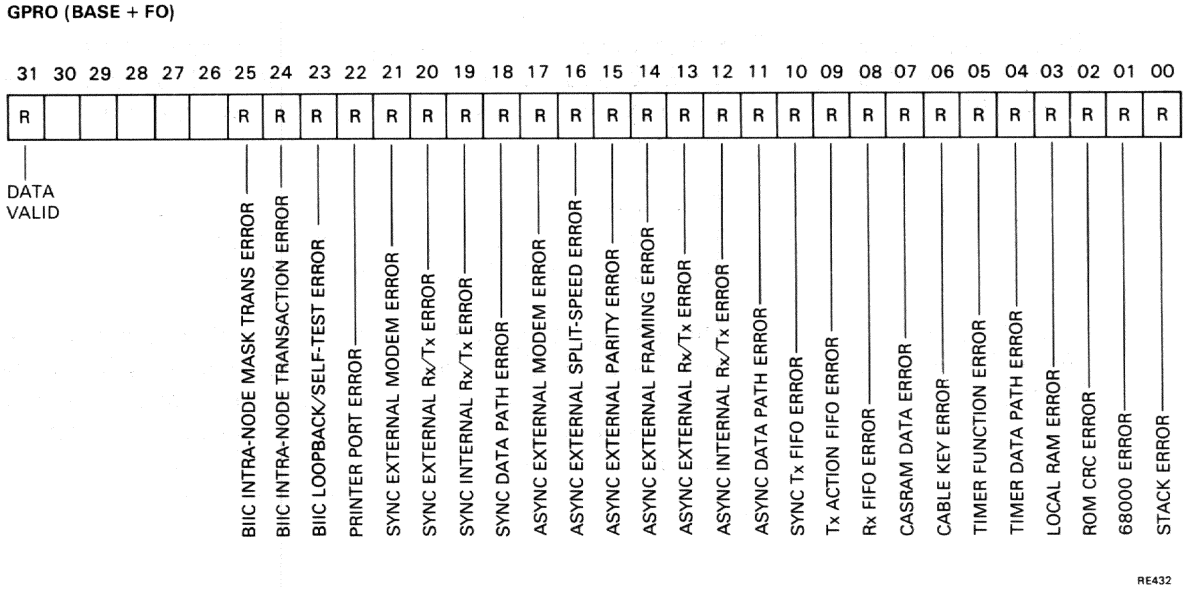
NOTE

When EX.VECTOR (bit <15>) is set, the other fields in this register do not need to be used. Since the DMB32 always has EX.VECTOR set, the host would not normally use this register. However, the information in bits <16:31> is still valid as described here.

Bit	Name	Description
<13:2>	VECTOR (Interrupt Vector)	The DMB32 does not use this field.
<15>	EX.VECTOR (External Vector) (R/W, Ec)	This bit is always set on the DMB32. This allows the firmware to specify the interrupt vector directly.

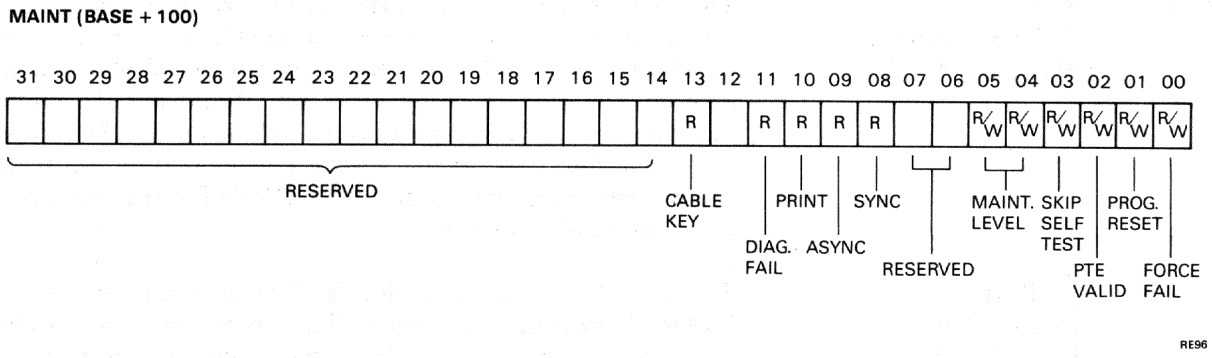
Bit	Name	Description
<19:16>	FORCE (INTR Force) (R/W, Ec, F)	The four bits correspond to the four interrupt levels. When a bit is set the BIIC generates an interrupt at the specified level.
<23:20>	SENT (INTR Sent) (W1C, Ec, F, *)	The four bits correspond to the four interrupt levels. When an INTR command has been successfully transmitted, the corresponding bit is set. The bit is cleared during an IDENT command following the detection of a level and Master ID match. Clearing the bit allows the interrupt to be resent if the node loses the IDENT arbitration or if the node wins but the vector transmission fails. The bit is cleared by removing the interrupt request.
<27:24>	INTRC (INTR Complete) (W1C, Ec, *)	The four bits correspond to the four interrupt levels. When the vector for an interrupt has been successfully transmitted, or if an INTR command sent under the control of this register is aborted, the corresponding bit is set. Removing the interrupt request clears the corresponding bit. While an INTRC bit is set, no further interrupts at that level are generated by this register, nor will this register respond to any IDENTS when the INTRC bit is set at IDENT level.
<31:28>	INTRAB (INTR Abort) (W1C, Ec, *)	The four bits correspond to the four interrupt levels. If an INTR command sent under the control of this register is aborted, the corresponding bit is set. These are status bits only and have no effect on the ability of the BIIC to send or respond to further INTR or IDENT commands.

3.3.9 General Purpose Register 0 (GPR0)



Bit	Name	Description
<31:0>	TSMR (R, Ec)	This register is used as the self-test status register TSR0. It provides for redundancy in DMB32 self-test error reporting should the RX FIFO be defective. After a self-test sequence, any errors that were found are reported here by setting the corresponding bit. Bit <31> is set to indicate that the contents of this register are valid. See also Section 4.3.2.

3.3.10 Maintenance Register (MAINT)

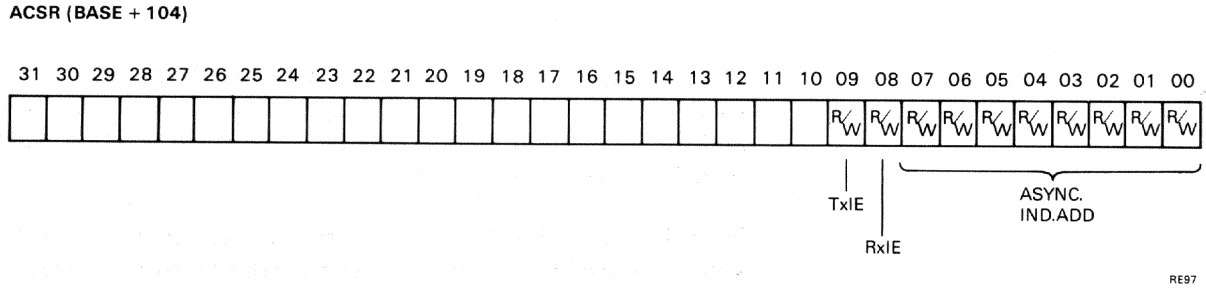


Bit	Name	Description
<0>	FORCE.FAIL (Force Failure) (R/W, D)	When this bit is set, the DMB32 will act as if the self-test has failed. This allows the self-test failure logic to be tested.
<1>	PROGRAMMED.RESET (Programmed Reset) (R/W, B)	Set by the host to request a programmed reset. This causes the DMB32 to run self-test and initialize itself. Programmed reset is the only way to use the Skip Self-Test and Forced Failure features. This bit is set by any reset operation. It stays set until the reset is complete. The host should not access any registers, other than MAINT and BIIC registers, while this bit is set.
<2>	PTE.VALID (Page Tables Valid) (R/W, A)	This bit is set by the host to indicate that it has set up the SPTE, SPTS, GPTE and GPTS registers. This bit should be cleared by the host when a programmed reset is requested, otherwise it may stay set (that is, it may not be automatically cleared by reset).

Bit	Name	Description
<3>	SKIP.SELF.TEST (Skip Self Test) (R/W, D)	When this bit is set, the DMB32 will bypass normal self-test operations. This results in a faster reset and allows diagnostics to test a board that fails self-test.
<5:4>	MAINT.LEVEL (Maintenance Level) (R/W, A)	These bits are used to determine the mode in which the DMB32 is to run. The bits have the following meaning: **<5> <4>** 0 0 Normal operation 0 1 Maintenance level 1 1 0 Maintenance level 2 1 1 Reserved In maintenance level 1, the Node window address space can be used to access the privileged registers contained on the board, and the host can directly access the CASRAM which is used to hold the registers and FIFOs. In maintenance level 2, the features at maintenance level 1 are still available, but in addition the DMB32's microprocessor is disabled. This allows the host to interrogate the microprocessor bus.
<8>	SYNC (Sync Line Present) (R, B)	This bit is set to indicate that an adapter cable or loopback connector is installed on the sync line at the distribution panel. This bit may be used by auto-configuration programs. If this bit is not set then the sync CSRs do not respond when they are addressed. This bit may be clear if certain self-test functions on the sync channels do not work.
<9>	ASYNC (Async Lines Present) (R, B)	This bit is set to indicate that the async channels are configured on the DMB32. This bit may be used by auto-configuration programs. If this bit is not set then the async CSRs do not respond when they are addressed. This bit may be clear if certain self-test functions on the async channels do not work.
<10>	PRINT (Printer Present) (R, B)	This bit is set to indicate that a suitable printer cable is connected to the the distribution panel. This bit may be used by auto-configuration programs. If this bit is not set then the printer CSRs do not respond when they are addressed. This bit may be clear if certain self-test functions on the printer interface do not work.
<11>	DIAG.FAIL (Diagnostic Error) (R, B)	If this bit remains set after PROGRAMMED.RESET has cleared then the self-test diagnostic has failed.

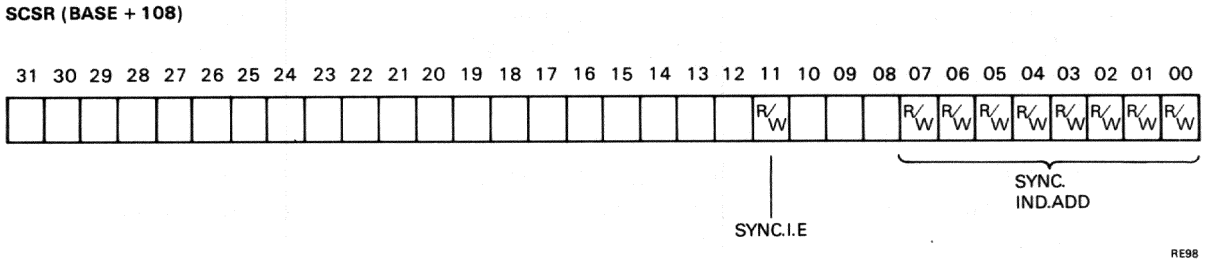
Bit	Name	Description
<13>	CABLE.KEY (Electrical Cable Key Signal Present) (R, B)	This bit is set to indicate that the self-test has detected the presence of the electrical-cable-key signal. This signal verifies that all six internal ribbon cables (17-00740-xx) are correctly connected between the module and the the distribution panel. This bit can be updated without a Reset occurring.
<31:14>	Not Used	These bits are reserved to DIGITAL.

3.3.11 Async Control And Status Register (ACSR)



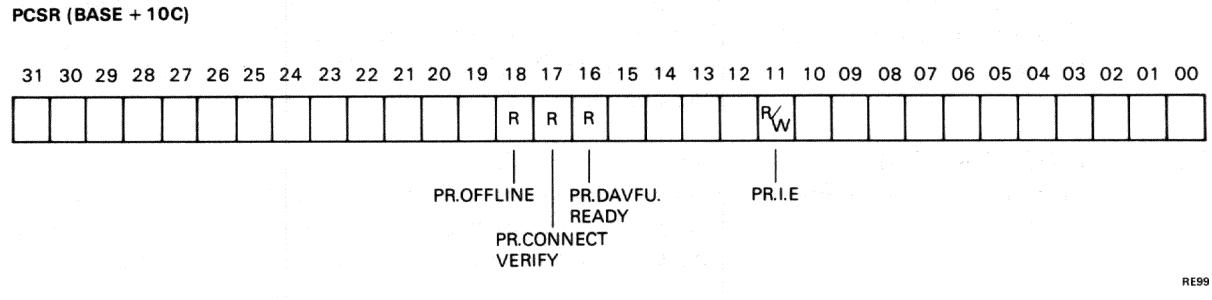
Bit	Name	Description
<7:0>	ASYNC.IND.ADD (Indirect Address Register Pointer) (R/W, B)	This byte should contain the channel number when accessing the following registers: LPR, FIFOSIZE, FIFODATA, LNCTRL, TBUFFAD, TBUFFCT, PREEMPT, LSTAT, FLOWC. Each of these registers controls a single channel. An access to any of them is indexed internally by the value of this field to access the register for the particular channel.
<8>	RX.I.E (Receive Interrupt Enable) (R/W, B)	When set, this bit allows the DMB32 to interrupt the CPU at the async interrupt vector when DATA.VALID (RBUF<31>) is set. Receive interrupts may be delayed by using the RX.TIMER, so that several characters can be processed during one interrupt.
<9>	TX.I.E (Transmit Interrupt Enable) (R/W, B)	When set, this bit allows interrupts to occur at the async interrupt vector when TX.ACT is set.

3.3.12 Sync Control And Status Register (SCSR)



Bit	Name	Description
<7:0>	SYNC.IND.ADD (Indirect Address Register Pointer) (R/W, B)	This byte defines the sync channel number when accessing the following registers: TBUFFAD1, TBUFFCT1, RBUFFAD1, RBUFFCT1, TLNCTRL1, RLNCTRL1, LPR1, LPR2, LPR3, BUFCTRL, TBUFFAD2, TBUFFCT2, RBUFFAD2, RBUFFCT2, TLNCTRL1, RLNCTRL2. Although the DMB32 has only one sync channel, and this byte is ignored, it is recommended that the driver use this register as if multiple sync channels are supported.
<11>	SYNC.I.E (Sync Interrupt Enable) (R/W, B)	When set, this bit allows the DMB32 to interrupt the CPU at the sync interrupt vector.

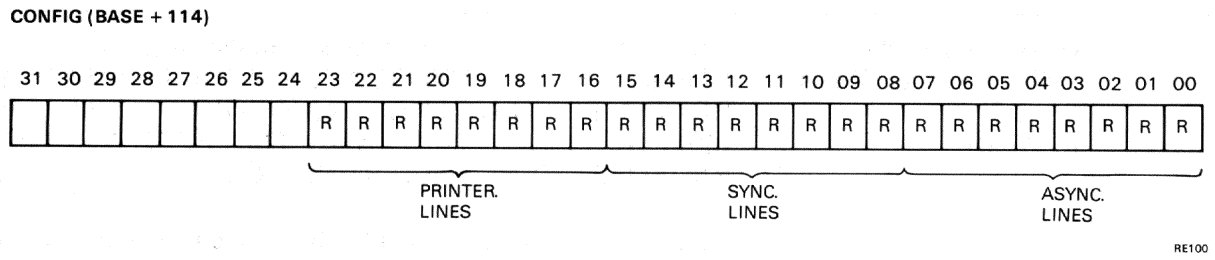
3.3.13 Printer Control And Status Register (PCSR)



Bit	Name	Description
<11>	PR.I.E (Printer Interrupt Vector) (R/W, B)	When set, this bit allows the DMB32 to interrupt the CPU at the printer interrupt vector.
<16>	PR.DAVFU.READY (DAVFU Ready) (R, B)	This bit indicates the state of the DAVFU Ready control line from the printer. When set the printer is ready to receive DAVFU characters.

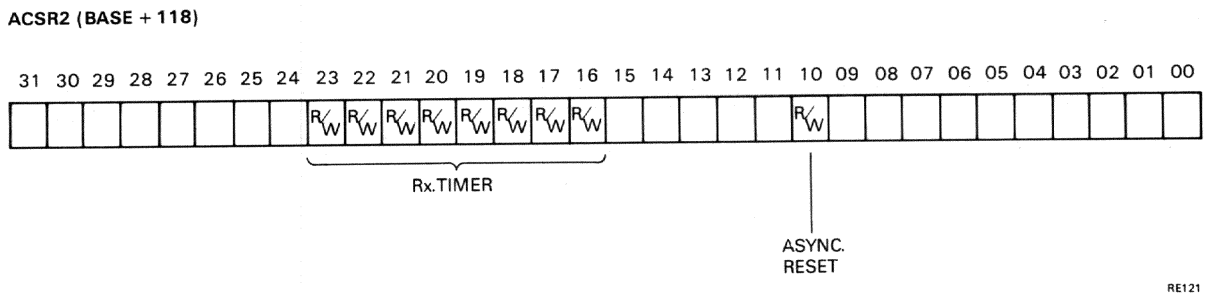
Bit	Name	Description
<17>	PR.CONNECT. READY (Connect Verify) (R, B)	This bit indicates the state of the Connect Verify control line from the printer. When set a printer is connected to the printer port.
<18>	PR.OFFLINE (Line Printer Error) (R, B)	This bit indicates the state of the Error control line from the printer. When set it indicates the printer is offline.

3.3.14 Device Configuration Register (CONFIG)



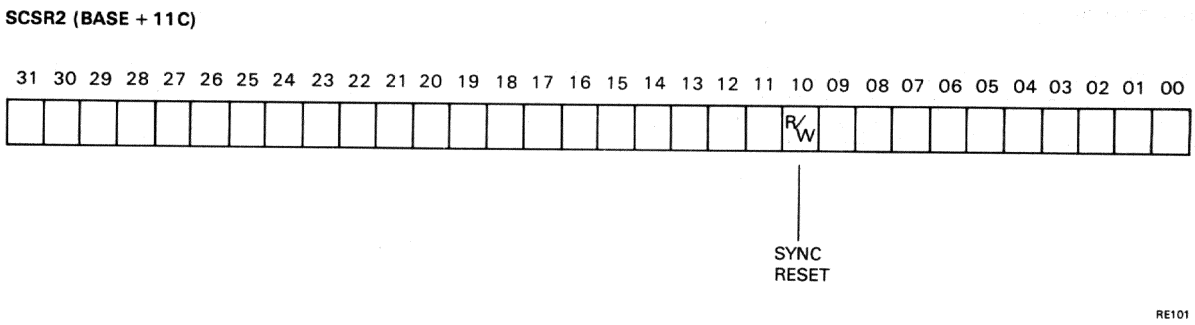
Bit	Name	Description
<7:0>	ASYNC.LINES (Number of Async Lines) (R, B)	This byte normally contains the value 08_{16}, the number of async channels supported by the DMB32. If the async channels are not configured (for example, self-test has failed), then this byte will read as zero.
<15:8>	SYNC.LINES (Number of Sync Lines) (R, B)	This byte normally contains the value 01_{16}, the number of sync channels supported by the DMB32. If there is no adapter cable connected, or if self-test has detected a hardware fault on the sync channel, then this byte will read as zero.
<23:16>	PRINTER.LINES (Number of Printer Ports) (R, B)	This byte normally contains the value 01_{16} the number of printer ports supported by the DMB32. If no printer cable is connected to the distribution panel, or if self-test has detected a hardware fault in the printer port, then this byte will read as zero.

3.3.15 Second Async Control And Status Register (ACSR2)



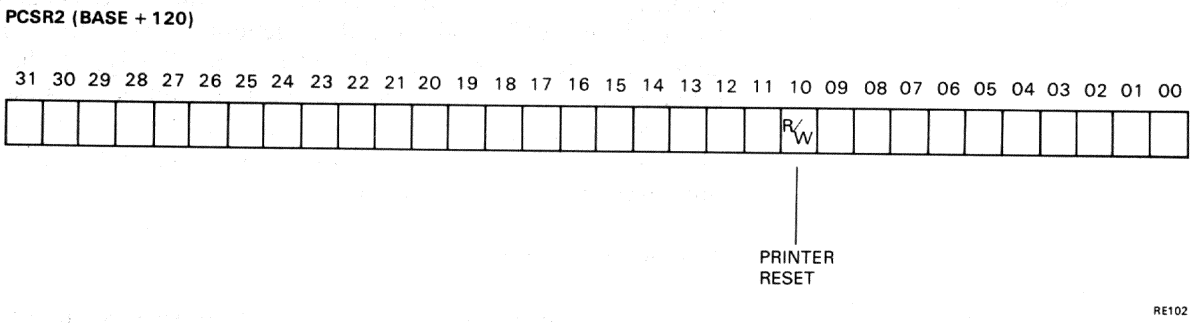
Bit	Name	Description
<10>	ASYNC.RESET (Async Port Reset) (R/W, C)	The host may set this bit to request a reset of the async channels. When the bit is set, the DMB32 will abort all async DMA operations and initialize all the async channels. When it has finished, it will clear the bit. After setting this bit, the host must not alter any registers associated with the async channels until this bit has been cleared. This process will not take longer than 10 ms. During the reset operation, activity on other channels may be affected (for example, overrun or underrun conditions may occur on the sync channels or the printer operation may be delayed). When the operation completes there will be no entries in RBUF or TBUF.
<23:16>	RX.TIMER (Receive Interrupt Delay Timer) (R/W, C)	This byte is used to delay a receive interrupt. This allows data to accumulate in the FIFO, and thus places less overhead on the host interrupt service routines. The delay may be selected from a number of values as indicated below. If the FIFO becomes so full that an XOFF code would be generated, then the delay is canceled and the interrupt signaled immediately. If the host empties the receive FIFO before the interrupt is signaled then the interrupt may or may not be requested. Any modem status-change entry causes an immediate interrupt, and any outstanding delay is canceled. Values for the timer: 0 — An infinite delay is used. A receive interrupt will only be given if the FIFO gets 3/4 full or a modem status change entry is put in the FIFO. 1 — No delay is used. The interrupt is delivered as soon as possible. 2 to 255 — Time in milliseconds.

3.3.16 Second Sync Control And Status Register (SCSR2)



Bit	Name	Description
<10>	SYNC.RESET (Sync Port Reset) (R/W)	This bit may be set by the host to request a reset of the sync channel. When this bit is set, the DMB32 will abort sync DMA operations and initialize the sync channel. When it has finished, it will clear this bit. Once the host has set this bit it must not alter any registers associated with the sync channel until the bit has been cleared. This process will not take longer than 10 ms. During the reset operation, activity on other channels may be affected (for example, overrun or underrun conditions may occur on the async channels or the printer operation may be delayed). When this operation completes there will be no entries in SBUF.

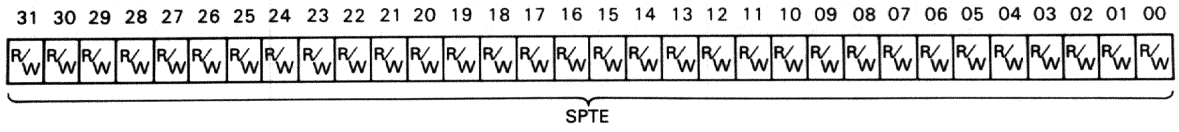
3.3.17 Second Printer Control And Status Register (PCSR2)



Bit	Name	Description
<10>	PRINTER.RESET (Printer Port Reset) (R/W)	This bit may be set by the host to request a reset of the printer port. When this bit is set, the DMB32 will abort DMA operations and initialize the printer port. When it has finished, it will clear this bit. Once the host has set this bit it must not alter any registers associated with the printer port until the bit has been cleared. This will not take longer than 10 ms. During the reset operation, activity on other channels may be affected (for example, overrun or underrun conditions may occur on the sync or async channels).

3.3.18 System Page Table Register (SPTE)

SPTE (BASE + 150)

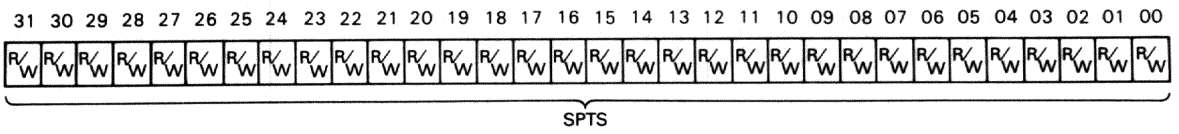


RE103

Bit	Name	Description
<31:0>	SPTE (System Page Table Register) (R/W)	This register is loaded with the physical address of the system page table. It is shared between all ports on the DMB32. It is recommended that each driver for each port load this register so that it will be valid whatever combination of ports are in use. The host should set PTE.VALID (MAINT<2>) only after loading the page table registers (this register together with SPTS, GPTE, and GPTS); if PTE.VALID is not set the contents of this register are not valid.

3.3.19 System Page Table Size Register (SPTS)

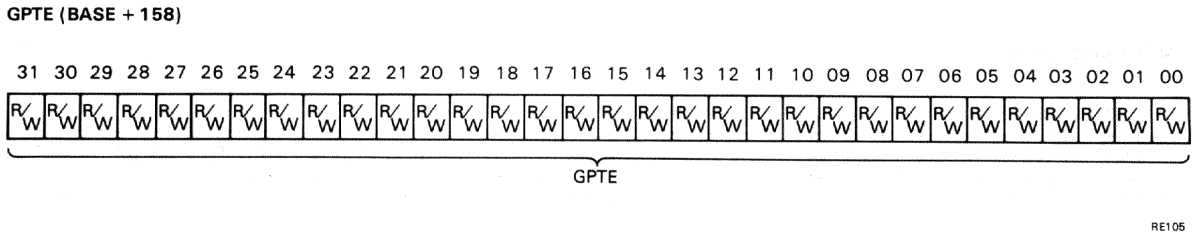
SPTS (BASE + 154)



RE104

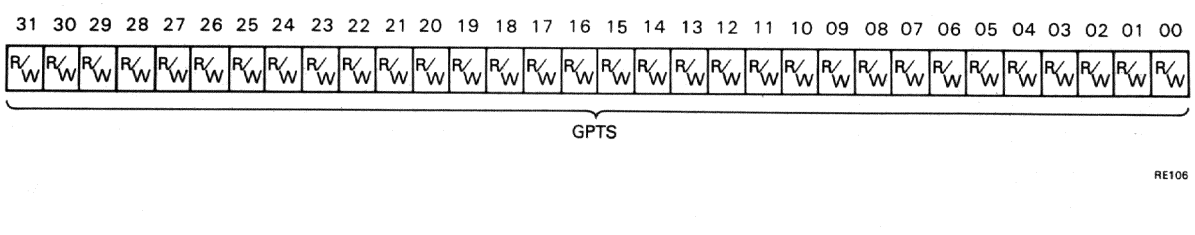
Bit	Name	Description
<31:0>	SPTS (System Page Table Size) (R/W)	This register is loaded with the number of entries in the system page table. It is shared between all ports on the DMB32. It is recommended that each driver for each port load this register so that it will be valid whatever combination of ports are in use. The host should set PTE.VALID (MAINT<2>) only after loading the page table registers (this register together with SPTE, GPTE, and GPTS); if PTE.VALID is not set the contents of this register are not valid.

3.3.20 Global Page Table Register (GPTE)



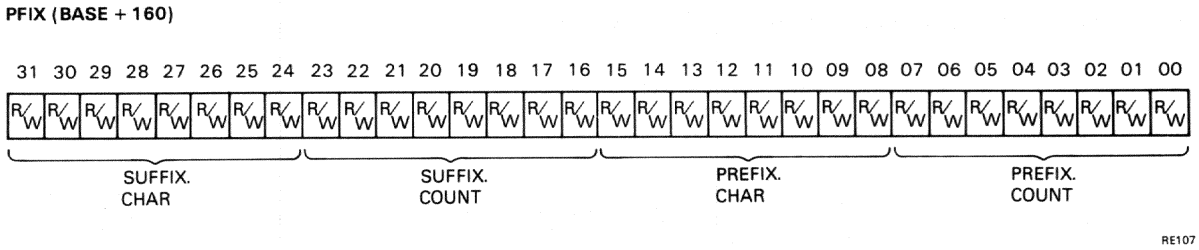
Bit	Name	Description
<31:0>	GPTE (Global Page Table Register) (R/W)	This register is loaded with the system virtual address of the global page table. It is shared between all ports on the DMB32. It is recommended that each driver for each port load this register so that it will be valid whatever combination of ports are in use. The host should set PTE.VALID (MAINT<2>) only after loading the page table registers (this register together with SPTE, SPTS, and GPTS); if PTE.VALID is not set the contents of this register are not valid.

3.3.21 Global Page Table Size Register (GPTS)



Bit	Name	Description
<31:0>	GPTS (Global Page Table Size) (R/W)	This register is loaded with the number of entries in the global page table. It is shared between all ports on the DMB32. It is recommended that each driver for each port load this register so that it will be valid whatever combination of ports are in use. The host should set PTE.VALID (MAINT<2>) only after loading the page table registers (this register together with SPTE, SPTS, and GPTE); if PTE.VALID is not set the contents of this register are not valid.

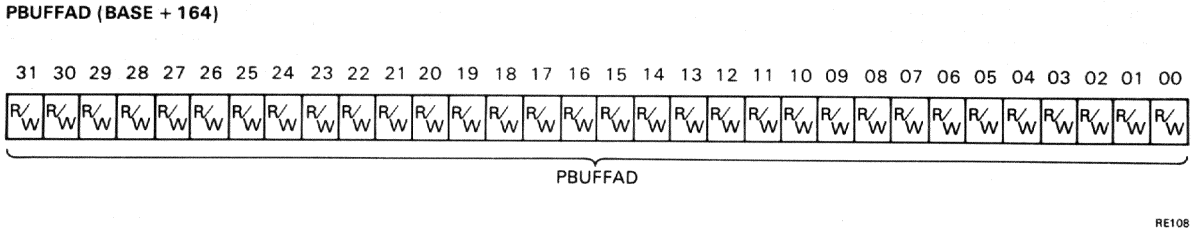
3.3.22 Printer Prefix/Suffix Control Register (PFIX)



Bit	Name	Description
<7:0>	PREFIX.COUNT (Prefix Count) (R/W)	Cleared by initialization. This byte is loaded by the host with the number of prefix characters to be inserted at the beginning of the buffer.
<15:8>	PREFIX.CHAR (Prefix Character) (R/W)	Cleared by initialization. This byte is loaded by the host with the type of prefix character to be inserted at the beginning of the buffer. If the prefix character count is zero then no prefix characters are sent. A zero value for this prefix character is interpreted as a "new line" character. A "new line" character is a carriage return followed by a line feed. The carriage return precedes the line feed only if the automatic carriage return insert bit (PR.AUTO.RETURN) is 1.
<23:16>	SUFFIX.COUNT (Suffix Count) (R/W)	Cleared by initialization. This byte is loaded by the host with the number of suffix characters to be inserted at the end of the buffer.

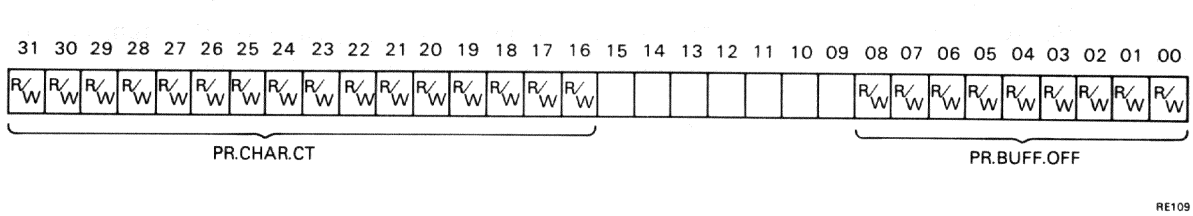
Bit	Name	Description
<31:24>	SUFFIX.CHAR (Suffix Character) (R/W)	Cleared by initialization. This byte is loaded by the host with the type of suffix character to be inserted at the end of the buffer. If the suffix character count is zero then no suffix characters are sent. A zero value for this suffix character is interpreted as a "new line" character. A "new line" character is a carriage return followed by a line feed. The carriage return precedes the line feed only if the automatic carriage return insert bit (PR.AUTO.RETURN) is 1.

3.3.23 Printer Buffer Address Register (PBUFFAD)



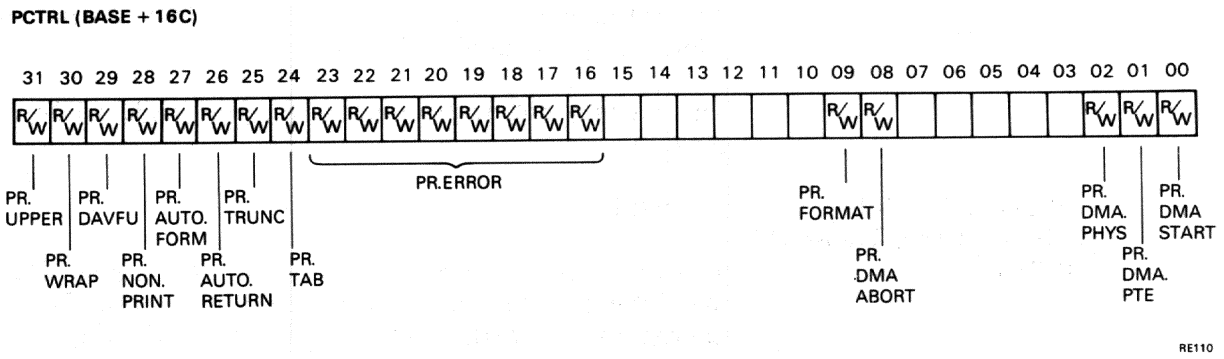
Bit	Name	Description
<31:0>	PBUFFAD (Printer Buffer Address) (R/W)	Cleared by initialization. This longword is loaded by the host with the start address of the buffer to be sent by DMA transfer. This register must not be written to between setting TX.DMA.START and the respective interrupt being returned. During this period, the contents of the register are undefined. The interpretation of this address is controlled by the DMA register.

3.3.24 Printer Buffer Count Register (PBUFFCT)



Bit	Name	Description
<8:0>	PR.BUFF.OFF (Printer Buffer Offset) (R/W)	These bits contain the offset of the first character of the buffer within the first page containing the buffer. This is only used when PR.DMA.PTE is set.
<31:16>	PR.CHAR.CT (Transmit DMA Character Count) (R/W)	Cleared by initialization. This word defines the length of the Transmit DMA buffer. It is initially loaded by the host and is updated by the DMB32 on termination of a transfer so that it contains the number of characters still to be sent. The number of characters is specified as a 16-bit unsigned integer. This register must not be written to between setting PR.DMA.START and the respective interrupt being returned. During this period, the contents of this register are undefined.

3.3.25 Printer Control Register (PCTRL)



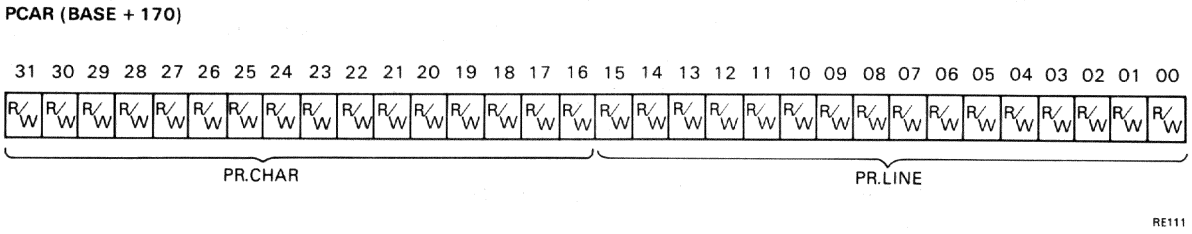
Bit	Name	Description
<0>	PR.DMA.START (Start a DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to start a printer port DMA transfer. Having set this bit, the host must not write to PBUFFCT or PBUFFAD until the interrupt has been received to say that this transfer has finished. This bit will be reset before the interrupt is returned.

Bit	Name	Description
<1>	PR.DMA.PTE (PTE Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is the address of the first PTE for the buffer. If the bit is clear, the DMB32 assumes that the address is the address of the first byte of the buffer. If the bit is set, the DMB32 assumes that the address is the address of a PTE which describes the page which contains the first byte of the buffer. This address must be longword aligned. The offset of the first byte is contained in PR.BUFF.OFF. This bit may not be modified by the host while a DMA is in progress.
<2>	PR.DMA.PHYS (Physical Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is a physical address. If this bit is clear, the DMB32 assumes the address is a system virtual address. This bit may not be modified by the host while a DMA is in progress.
<8>	PR.DMA.ABORT (Abort a DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to abort a DMA transfer. The host may set it at any time. If this bit is set while a print operation is in progress the DMB32 will abort the operation and generate an interrupt. (It may not be possible to determine at what point the operation was aborted; for example, if prefix or suffix characters were being transmitted when the abort was detected).
<9>	PR.FORMAT (Format Control) (R/W)	Cleared by Master Reset. If this bit is set, the DMB32 examines the characters being transferred from the DMA buffer to the printer. Characters may be acted upon as indicated by the formatting control bits. If this bit is clear, all characters are transferred without being examined. The DMB32 is unable to keep track of the carriage position, and the PR.AUTO.RETURN, PR.AUTO.FORM, PR.NON.PRINT, PR.DAVFU, PR.WRAP, PR.UPPER, PR.TAB and PR.TRUNC bits have no effect.

Bit	Name	Description
<23:16>	PR.ERROR (Error Code) (R/W)	Cleared by initialization. If it contains a value other than zero, this field indicates that errors were detected during a DMA operation. Error codes are described in Section 3.4.8.
<24>	PR.TAB (Tab Expansion) (R/W)	Cleared by initialization. If this bit is set the DMB32 will convert any TAB characters to the appropriate number of spaces while formatting output data. Tabs are set every eighth character position.
<25>	PR.TRUNC (Truncation of Data) (R/W)	Cleared by initialization. When this bit is set the DMB32 will truncate overlong lines while formatting output data.
<26>	PR.AUTO.RETURN (Auto Carriage Return Insert) (R/W)	Cleared by initialization. If this bit is set, the DMB32 will insert carriage returns before any form feed or line feed in the data stream that is not preceded by a carriage return. It will also strip out any redundant carriage returns.
<27>	PR.AUTO.FORM (Auto Form Feed to Line Feed Convert) (R/W)	Cleared by initialization. If this bit is set the DMB32 will convert a form feed to the appropriate number of line feeds in order to position the paper to the top of the form. The page size is specified in PR. The DMB32 assumes that the paper is positioned at top-of-form when the printer port is initialized.
<28>	PR.NON.PRINT (Non-Printing Character Accept) (R/W)	Cleared by initialization. This bit allows non-printing characters to be included in the data stream. Non-printing characters are those with the MSB set, and also unknown control codes. If formatting is enabled and this bit is clear, then non-printing characters are discarded. If formatting is enabled and this bit is set, then these characters are sent to the printer. Unknown control codes are assumed to have no effect on the carriage position. Other non-printing characters are assumed to take one print position.

Bit	Name	Description
<29>	PR.DAVFU (DAVFU) (R/W)	Cleared by initialization. This bit allows DAVFU control codes (200_8 to 237_8) to be transferred to the printer. The DMB32 keeps track of the carriage position as it is affected by these codes. If this bit is not set then these codes are considered non-printing codes and are dealt with as indicated by the PR.NON.PRINT bit.
<30>	PR.WRAP (Line Wrap) (R/W)	Cleared by initialization. If this bit is set, a new-line sequence is sent to the printer if the current character would be printed at a position beyond the end of the line, as indicated by the line size in PR.WIDTH. If this bit is clear, the rest of the line will either be sent to the printer or truncated, as specified by the PR.TRUNC bit.
<31>	PR.UPPER (Convert to Upper Case) (R/W)	Cleared by initialization. This bit causes the DMB32 to translate all lower-case characters to the corresponding upper-case character (this includes some non-alphabetic characters that do not appear on upper-case-only printers).

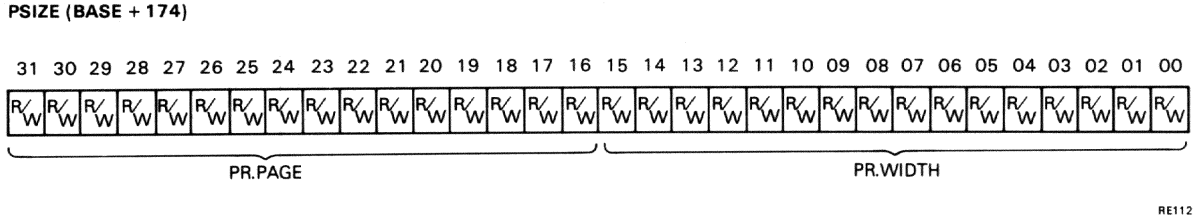
3.3.26 Printer Carriage Counter Register (PCAR)



Bit	Name	Description
<15:0>	PR.LINE (Lines Printed) (R/W)	Cleared by initialization. This word contains the number of lines of paper that have been used during the previous DMA (that is, the number of line feed characters and the number of line feeds that a form feed would have been converted to). It is only valid when PR.DMA.START is zero. It is not valid if DAVFU is set, or if formatting is not enabled.

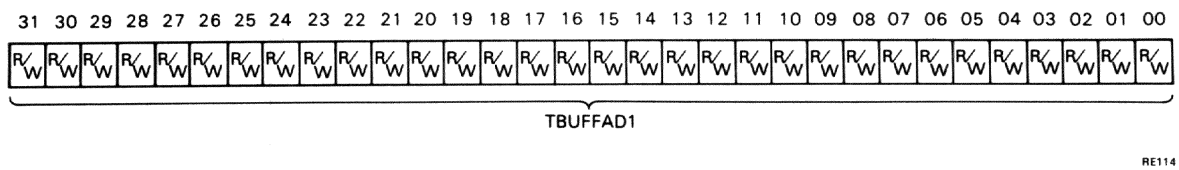
Bit	Name	Description
<31:16>	PR.CHAR (Characters Transmitted) (R/W)	Cleared by initialization. This word indicates the number of characters transferred to the line printer during the previous DMA operation. This includes all prefix and suffix characters and all inserted characters. It is only valid when PR.DMA.START is zero.

3.3.27 Printer Page Size Descriptor Register (PSIZE)



Bit	Name	Description
<15:0>	PR.WIDTH (Line Width) (R/W)	Cleared by initialization. This word contains the number of printing positions on a line. Zero indicates no limit to the number of printing positions. It is used to determine when a new-line should be inserted when auto-wrap operation is selected.
<31:16>	PR (Page Size) (R/W)	Cleared by initialization. This word contains the number of lines which may be printed on a page. It is used when converting form feeds to line feeds and when accounting for the number of lines used during a DMA.

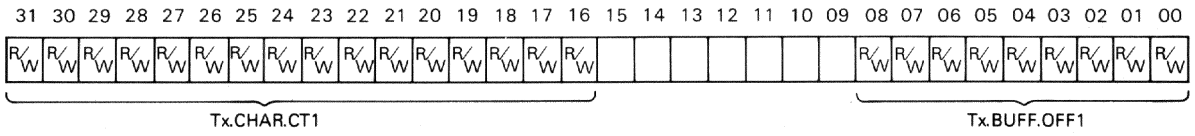
3.3.28 Sync Transmit Buffer 1 Address Register (TBUFFAD1)



Bit	Name	Description
<31:0>	TBUFFAD1 (Buffer Address) (R/W)	Cleared by initialization. This longword is loaded by the host with the start address of the buffer to be sent on the sync channel by DMA transfer.

3.3.29 Sync Transmit Buffer 1 Count/Offset Register (TBUFFCT1)

TBUFFCT1 (BASE + 184 (MODIFIED BY SYNC.IND.ADD))

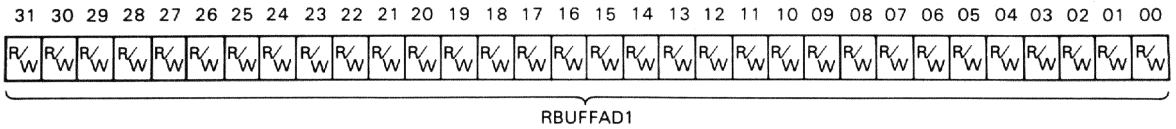


RE115

Bit	Name	Description
<8:0>	TX.BUFF.OFF1 (Transmit Buffer Offset) (R/W)	These bits contain the offset of the first character of the buffer within the first page containing the buffer. This is only used when TX.DMA.PTE is set.
<31:16>	TX.CHAR.CT1 (Transmit DMA Character Count) (R/W)	Cleared by initialization. These bits define the length of the Transmit DMA buffer for the selected channel. They are initially loaded by the host and are updated by the DMB32 after a transfer so that they contain the number of characters still to be sent. The number of characters is specified as a 16-bit unsigned integer.

3.3.30 Sync Receive Buffer 1 Address Register (RBUFFAD1)

RBUFFAD1 (BASE + 188 (MODIFIED BY SYNC.IND.ADD))

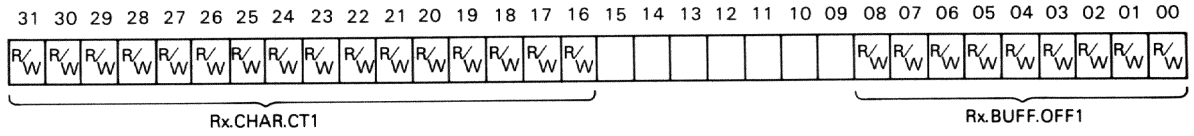


RE116

Bit	Name	Description
<31:0>	RBUFFAD1 (Buffer Address) (R/W)	Cleared by initialization. This longword is loaded by the host, with the start address of the buffer to receive data from the sync channel by DMA transfer.

3.3.31 Sync Receive Buffer 1 Count/Offset Register (RBUFFCT1)

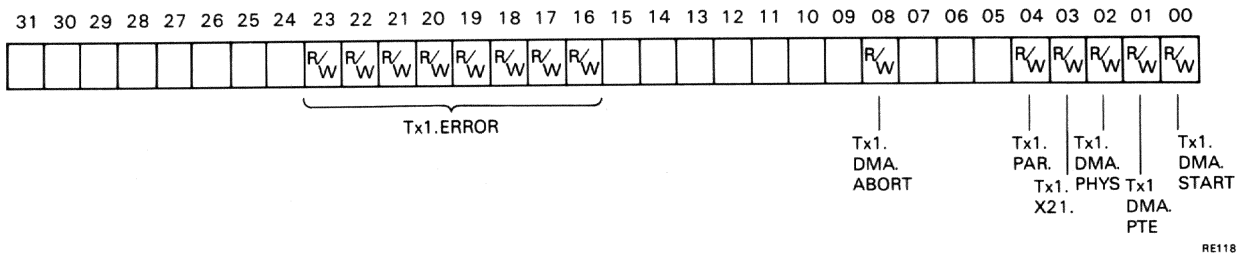
RBUFFCT1 (BASE + 18C (MODIFIED BY SYNC.IND.ADD))



Bit	Name	Description
<8:0>	RX.BUFF.OFF1 (Receive Buffer Offset) (R/W)	These bits contain the offset of the first character of the buffer within the first page containing the buffer. This is only used when TX.DMA.PTE is set.
<31:16>	RX.CHAR.CT1 (Receive DMA Character Count) (R/W)	Cleared by initialization. These bits define the length of the Receive DMA buffer for the selected channel. They are initially loaded by the host and are updated by the DMB32 after a transfer so that they contain the number of characters actually received. The number of characters is specified as a 16-bit unsigned integer.

3.3.32 Sync Transmit Buffer 1 Control Register (TLNCTRL1)

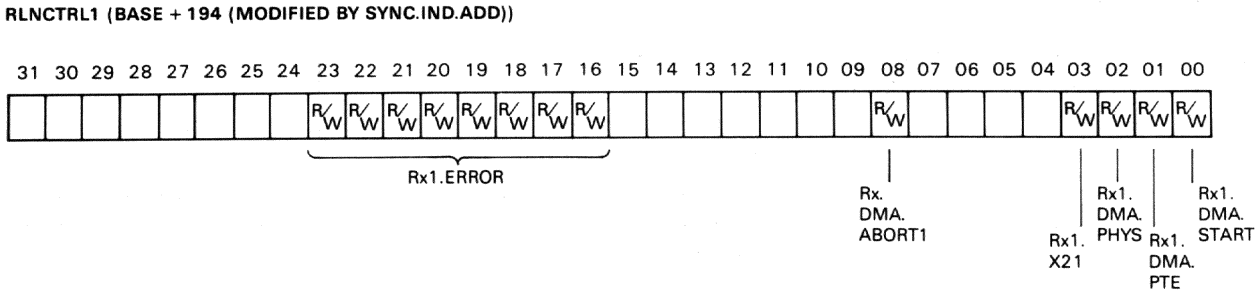
TLNCTRL1 (BASE + 190 (MODIFIED BY SYNC.IND.ADD))



Bit	Name	Description
<0>	TX1.DMA.START (Start a DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to start a sync port DMA transfer from buffer 1. Having set this bit, the host must not write to TBUFFCT1 or TBUFFAD1 until this bit is cleared by the DMB32 at the end of transmission.
<1>	TX1.DMA.PTE (PTE Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is the address of the first PTE for the buffer. If this bit is clear, the DMB32 assumes that the address is the address of the first byte of the buffer. If this bit is set, the DMB32 assumes that the address is the address of a PTE describing the page containing the first byte of the buffer. This address must be longword aligned and the offset of the first byte is contained in TX1.BUFF.OFF. This bit may not be modified by the host while a DMA is in progress.
<2>	TX1.DMA.PHYS (Physical Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is a physical address. If this bit is clear, the DMB32 assumes the the address is a system virtual address. This bit may not be modified by the host while a DMA is in progress.
<3>	TX1.X21 (R/W)	Reserved to DIGITAL.
<4>	TX1.PAR (R/W)	Reserved to DIGITAL.
<8>	TX1.DMA.ABORT (Transmit DMA Abort) (R/W)	Cleared by initialization. This bit is set by the host, to abort transmission of data from buffer 1 of the sync channel. On seeing this bit set, the DMB32 stops transmitting data on this channel. If a DMA was in progress, it updates the DMA address/count registers and generates an interrupt. This bit should be clear before a transfer is started otherwise the new transfer will be aborted without transmitting any characters.

Bit	Name	Description
<23:16>	TX1.ERROR (Transmitter Error Bits) (R/W)	Cleared by initialization. If this byte contains a value other than zero, this indicates that errors were detected during an operation on the transmitter. Error codes are described in Section 3.4.8.

3.3.33 Sync Receive Buffer 1 Control Register (RLNCTRL1)

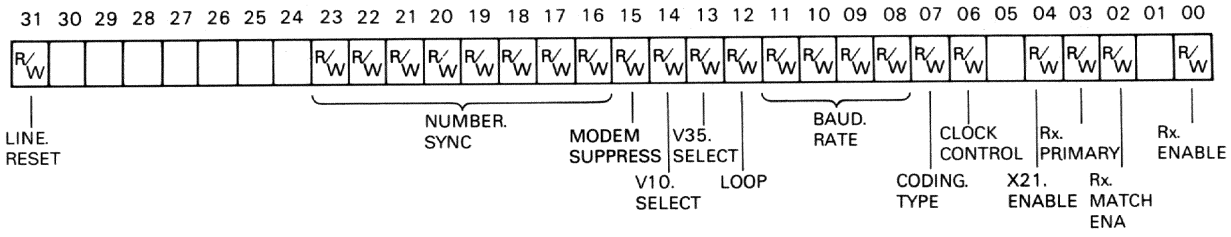


Bit	Name	Description
<0>	RX1.DMA.START (Start a DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to start a sync port DMA transfer to buffer 1. Having set this bit, the host must not write to RBUFFCT1 or RBUFFAD1 until this bit is cleared by the DMB32 at the end of transmission.
<1>	RX1.DMA.PTE (PTE Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is the address of the first PTE for the buffer. If this bit is clear, the DMB32 assumes that the address is the address of the first byte of the buffer. If this bit is set, the DMB32 assumes that the address is the address of a PTE which describes the page which contains the first byte of the buffer. This address must be longword aligned. The offset of the first byte is contained in RX1.BUFF.OFF. This bit may not be modified by the host while a DMA is in progress.

Bit	Name	Description
<2>	RX1.DMA.PHYS (Physical Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is a physical address. If this bit is clear, the DMB32 assumes the the address is a system virtual address. This bit must not be modified by the host while a DMA is in progress.
<3>	RX1.X21 (R/W)	Reserved to DIGITAL.
<8>	RX.DMA.ABORT1 (Receiver DMA Abort) (R/W)	Cleared by initialization. This bit is set by the host, to abort reception of data to buffer 1 of the sync channel. On seeing this bit set, the DMB32 stops receiving data on this channel. If a DMA was in progress it updates the DMA address/count registers and generates an interrupt. This bit should be clear before a transfer is started otherwise the new transfer will be aborted without receiving any characters.
<23:16>	RX1.ERROR (Receiver Error Bits) (R/W)	Cleared by initialization. If this byte contains any value other than zero, this field indicates that errors were detected during an operation on the receiver. Error codes are described in Section 3.4.8.

3.3.34 Sync Line Parameters Register 1 (LPR1)

LPR1 (BASE + 198 (MODIFIED BY SYNC.IND.ADD))



Bit	Name	Description
<0>	RX.ENABLE (Receiver Enable) (R/W)	When set, this bit enables processing of the receiver serial input data.
<2>	RX.MATCH.ENA (Receive Match Character Enable) (R/W)	This bit is used by the GEN BYTE protocol. It enables the matching of characters using the MATCH character.
<3>	RX.PRIMARY (Primary/Secondary Station) (R/W)	This bit controls whether the device acts as a primary station or as a secondary station. When set the DMB32 acts as a secondary station, when clear as a primary station.
<4>	X21.ENABLE (R/W)	Reserved to DIGITAL.
<6>	CLOCK.CONTROL (Clock Control Bits) (R/W)	When this bit is set, it indicates that the internal clock should be used, rather than the DTE clock from the interface. This would normally be used for internal loop.
<7>	CODING.TYPE (Data Coding Type) (R/W)	If this bit is set then the data transmitted and received will be encoded using NRZI encoding. If this bit is clear then the data will be transmitted and received normally.

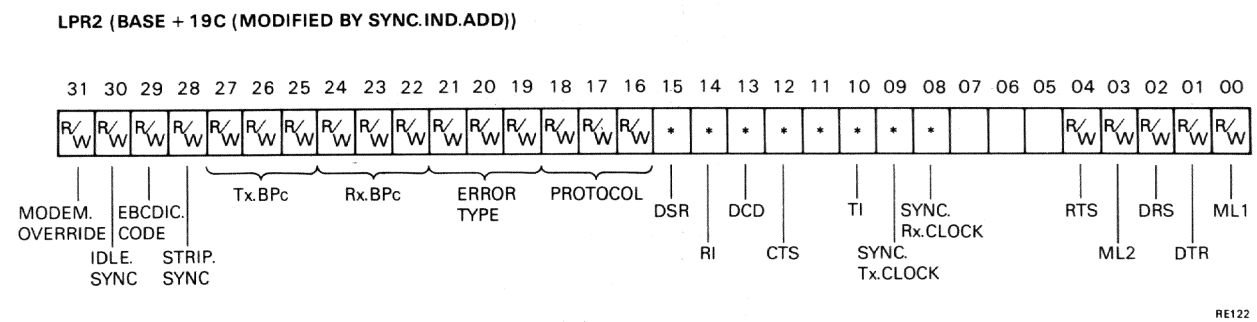
Bit	Name	Description
<11:8>	BAUD.RATE<3:0> (Internal Baud Rate Generator) (R/W)	This field selects the speed of the internal baud rate generator. (The baud rate clock is passed to the DTE clock signal of the sync connector. It is also used when the clock control bit indicates that the internal clock should be used). Possible values are: 0 - 600 1 - 1200 2 - 1800 3 - 2000 4 - 2400 5 - 4800 6 - 9600 7 - 19200 8 - 48000 9 - 56000 10 - 64000 11 - reserved 12 - reserved 13 - reserved 14 - no clock (DTE clock off) 15 - maximum sensible speed 'Maximum sensible speed' is the highest speed that the device supports for the selected protocol. These speeds are given in the following table (speeds are given in bits/s).

	Electrical Interface Standard			
Protocol	**RS-232**	**RS-423**	**RS-422**	**V.35**
DDCMP	19200	19200	19200	19200
HDLC	19200	64000	64000	48000
BISYNC	9600	9600	9600	9600
GEN BYTE	9600	9600	9600	9600

<12>	LOOP (Maintenance Loopback) (R/W)	Cleared by master reset. This bit controls the internal loopback of the sync channel. If it is set then data is internally looped back.
<13>	V35.SELECT (V.35 Select) (R/W)	This bit is used by the host when some maintenance mode connectors are being used. When the bit is set, the DMB32 uses the V.35 receivers instead of the V.11 receivers. This bit is only used for maintenance; in normal use the type of sync cable connected will select the appropriate receivers.

Bit	Name	Description
<14>	V10.SELECT (V.10 Select) (R/W)	This bit is used by the host when some maintenance mode connectors are being used. When this bit is set, the DMB32 uses the V.10 receivers instead of the V.11 receivers. This bit is only used for maintenance; in normal use the type of sync cable connected will select the appropriate receivers.
<15>	MODEM.SUPPRESS (Inhibit Modem FIFO Entries) (R/W)	This bit is set by the host to suppress the normal action of putting modem status change entries into the FIFO.
<23:16>	NUMBER.SYNC (Number of Sync Characters) (R/W)	This field indicates the number of sync characters that should be transmitted before a message. This field should normally be set to 3 for BISYNC and 2 for DDCMP.
<31>	LINE.RESET (Line Reset Request) (R/W)	If this bit is set then the DMB32 will abort all DMA operations and initialize the channel. When it has finished, it will clear this bit. Once the host has set this bit it must not alter any registers associated with this channel until this bit has been cleared. This will not take longer than 10 ms. During the reset operation activity on other channels may be affected (for example, overrun or underrun conditions may occur on other sync channels and received characters may be lost on async channels). If there are DMA operations in progress on the channel then the FIFO entries for these may be present in SBUF, or they may be lost.

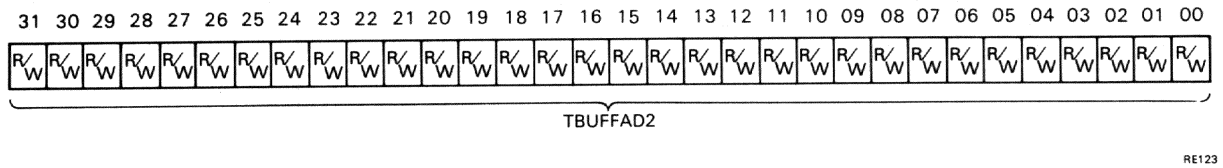
3.3.35 Sync Line Parameters Register 2 (LPR2)



Bit	Name	Description
<4:0>	(Modem Control Outputs) (R/W)	These bits are used to directly control the modem status outputs. Each bit corresponds to one control line as follows: **Bit** **Control Line** 0 ML1 (Modem Loop (Local)) 1 DTR (Data Terminal Ready) 2 DRS (Data Rate Select) 3 ML2 (Modem Loop (Remote)) 4 RTS (Request To Send)
<15:8>	(Modem Status Inputs) (R)	These bits are used to monitor the received modem status input. Each bit corresponds to one line as follows: **Bit** **Status Line** 8 Rx Clock running 9 Tx Clock running 10 TI (Test Indicator) 11 (Not used) 12 CTS (Clear To Send) 13 DCD (Data Carrier Detect) 14 RI (Ring Indicator) 15 DSR (Data Set)
<18:16>	PROTOCOL (Protocol Type) (R/W)	0 - DDCMP 1 - SDLC 2 - HDLC 3 - IBM BISYNC 4 - Spare 5 - Spare 6 - Spare 7 - GEN BYTE
<21:19>	ERROR.TYPE (Error Control Type) (R/W)	0 - CRC-CCITT preset to 1s 1 - CRC-CCITT preset to 0s 2 - LRC/VRC odd 3 - CRC-16 4 - LRC odd 5 - LRC even 6 - LRC/VRC even 7 - no error control
<24:22>	RX.BPC (Number of Receive Bits per Character) (R/W)	0 - 8 bits 6 - 6 bits 7 - 7 bits GEN BYTE only The DMB32 does not support character sizes less than 6 bits per character.

Bit	Name	Description
<27:25>	TX.BPC (Number of Transmit Bits per Character) (R/W)	0 – 8 bits 6 – 6 bits 7 – 7 bits GEN BYTE only The DMB32 does not support character sizes less than 6 bits per character.
<28>	STRIP.SYNC (Strip Sync) (R/W)	This bit indicates whether sync characters should be stripped for the GEN BYTE protocol.
<29>	EBCDIC.CODE (Character Code) (R/W)	This bit indicates whether the character code used for IBM BISYNC is EBCDIC (bit <29> set) or ASCII (bit <29> clear).
<30>	IDLE.SYNC (Idle Sync) (R/W)	This bit indicates whether the DMB32 should send sync characters while idle (if set) or MARK (if clear).
<31>	MODEM.OVERRIDE (Modem Control Override)	If this bit is set, the DMB32 will ignore modem control signals. If the bit is clear, the DMB32 will not allow transmission or reception to proceed if the modem control bits are set incorrectly (see Section 3.4.11).

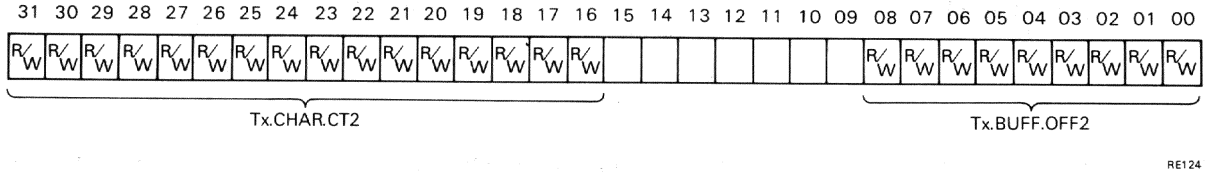
3.3.36 Sync Transmit Buffer 2 Address Register (TBUFFAD2)



Bit	Name	Description
<31:0>	TBUFFAD2 (Buffer Address) (R/W)	Cleared by initialization. This longword is loaded by the host with the start address of the buffer to be sent on the sync channel by DMA transfer.

3.3.37 Sync Transmit Buffer 2 Count/Offset Register (TBUFFCT2)

TBUFFCT2 (BASE + 1A4 (MODIFIED BY SYNC.IND.ADD))

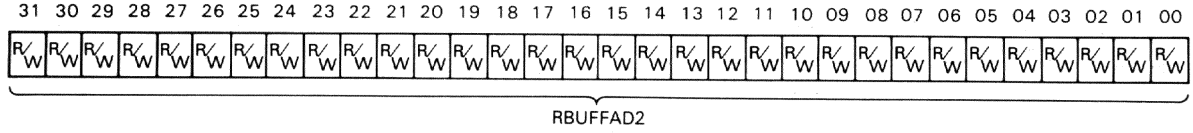


RE124

Bit	Name	Description
<8:0>	TX.BUFF.OFF2 (Transmit Buffer Offset) (R/W)	These bits contain the offset of the first character of the buffer within the first page containing the buffer. This is only used when TX.DMA.PTE is set.
<31:16>	TX.CHAR.CT2 (Transmit DMA Character Count) (R/W)	Cleared by initialization. These bits define the length of the Transmit DMA buffer for the selected channel. They are initially loaded by the host and are updated by the DMB32 after a transfer so that they contain the number of characters still to be sent. The number of characters is specified as a 16-bit unsigned integer.

3.3.38 Sync Receive Buffer 2 Address Register (RBUFFAD2)

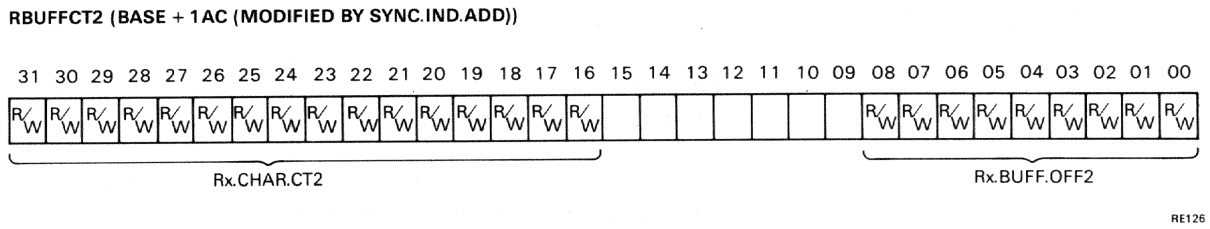
RBUFFAD2 (BASE + 1A8 (MODIFIED BY SYNC.IND.ADD))



RE125

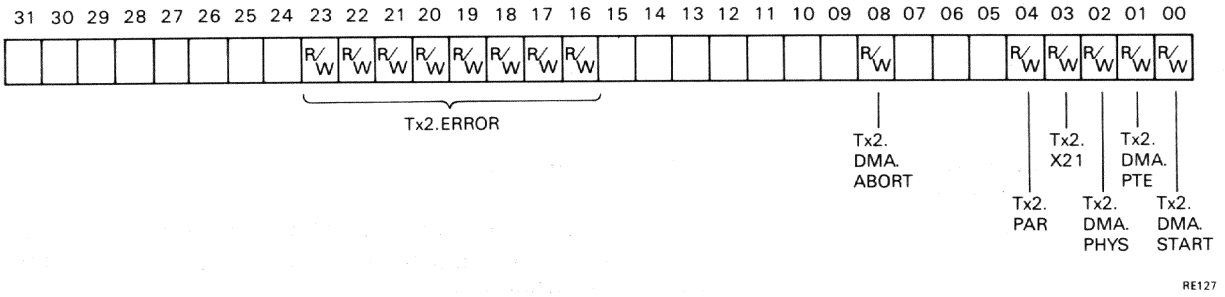
Bit	Name	Description
<31:0>	RBUFFAD2 (Buffer Address) (R/W)	Cleared by initialization. This longword is loaded by the host, with the start address of the buffer to receive data from the sync channel by DMA transfer.

3.3.39 Sync Receive Buffer 2 Count/Offset Register (RBUFFCT2)



Bit	Name	Description
<8:0>	RX.BUFF.OFF2 (Receive Buffer Offset) (R/W)	These bits contain the offset of the first character of the buffer within the first page containing the buffer. This is only used when TX.DMA.PTE is set.
<31:16>	RX.CHAR.CT2 (Receive DMA Character Count) (R/W)	Cleared by initialization. These bits define the length of the Receive DMA buffer for the selected channel. They are initially loaded by the host and are updated by the DMB32 after a transfer so that they contain the number of characters actually received. The number of characters is specified as a 16-bit unsigned integer.

3.3.40 Sync Transmit Buffer 2 Control Register (TLNCTRL2)

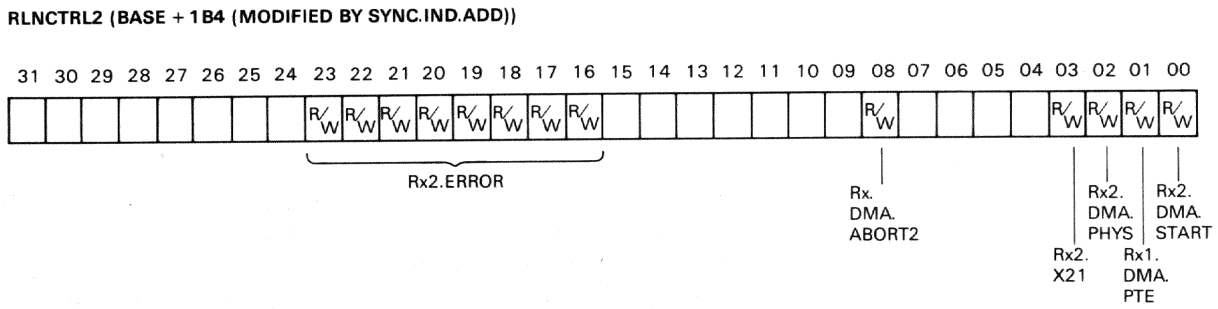


Bit	Name	Description
<0>	TX2.DMA.START (Start a DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to start a sync port DMA transfer from buffer 2. Having set this bit, the host must not write to TBUFFCT2 or TBUFFAD2 until this bit is cleared by the DMB32 at the end of transmission.

Bit	Name	Description
<1>	TX2.DMA.PTE (PTE Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is the address of the first PTE for the buffer. If this bit is clear, the DMB32 assumes that the address is the address of the first byte of the buffer. If this bit is set, the DMB32 assumes that the address is the address of a PTE which describes the page which contains the first byte of the buffer. This address must be longword aligned and the offset of the first byte is contained in TX2.BUFF.OFF. This bit may not be modified by the host while a DMA is in progress.
<2>	TX2.DMA.PHYS (Physical Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is a physical address. If this bit is clear, the DMB32 assumes the the address is a system virtual address. This bit may not be modified by the host while a DMA is in progress.
<3>	TX2.X21 (R/W)	Reserved to DIGITAL.
<4>	TX2.PAR (R/W)	Reserved to DIGITAL.
<8>	TX2.DMA.ABORT (Transmit DMA Abort) (R/W)	Cleared by initialization. This bit is set by the host, to abort transmission of data on buffer 1 of the sync channel. On seeing this bit set, the DMB32 stops transmitting data on this channel. If a DMA was in progress, it updates the DMA address/count registers and generates an interrupt. This bit should be clear before a transfer is started otherwise the new transfer will be aborted without transmitting any characters.

Bit	Name	Description
<23:16>	TX2.ERROR (Transmitter Error Bits) (R/W)	Cleared by initialization. If this byte contains a value other than zero, this indicates that errors were detected during an operation on the transmitter. Error codes are described in Section 3.4.8.

3.3.41 Sync Receive Buffer 2 Control Register (RLNCTRL2)

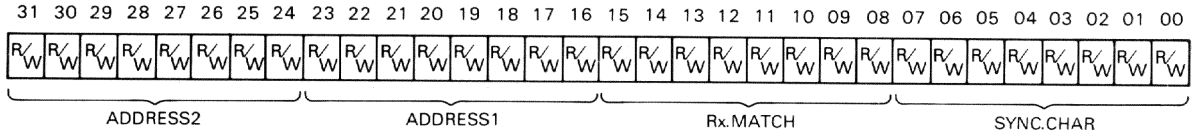


Bit	Name	Description
<0>	RX2.DMA.START (Start a DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to start a sync port DMA transfer to buffer 2. Having set this bit, the host must not write to RBUFFCT1 or RBUFFAD1 until this bit is cleared by the DMB32 at the end of transmission.
<1>	RX2.DMA.PTE (PTE Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is the address of the first PTE for the buffer. If this bit is clear, the DMB32 assumes that the address is the address of the first byte of the buffer. If this bit is set, the DMB32 assumes that the address is the address of a PTE describing the page containing the first byte of the buffer. This address must be longword aligned. The offset of the first byte is contained in RX2.BUFF.OFF. This bit may not be modified by the host while a DMA is in progress.

Bit	Name	Description
<2>	RX2.DMA.PHYS (Physical Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is a physical address. If this bit is clear, the DMB32 assumes the the address is a system virtual address. This bit must not be modified by the host while a DMA is in progress.
<3>	RX2.X21 (R/W)	Reserved to DIGITAL.
<8>	RX.DMA.ABORT2 (Receiver DMA Abort) (R/W)	Cleared by initialization. This bit is set by the host, to abort reception of data on buffer 1 of the sync channel. On seeing this bit set, the DMB32 stops receiving data on this channel. If a DMA was in progress it updates the DMA address/count registers and generates an interrupt. This bit should be clear before a transfer is started otherwise the new transfer will be aborted without receiving any characters.
<23:16>	RX2.ERROR (Receiver Error Bits) (R/W)	Cleared by initialization. If this byte contains any value other than zero, this field indicates that errors were detected during an operation on the receiver. Error codes are described in Section 3.4.8.

3.3.42 Sync Line Parameters Register 3 (LPR3)

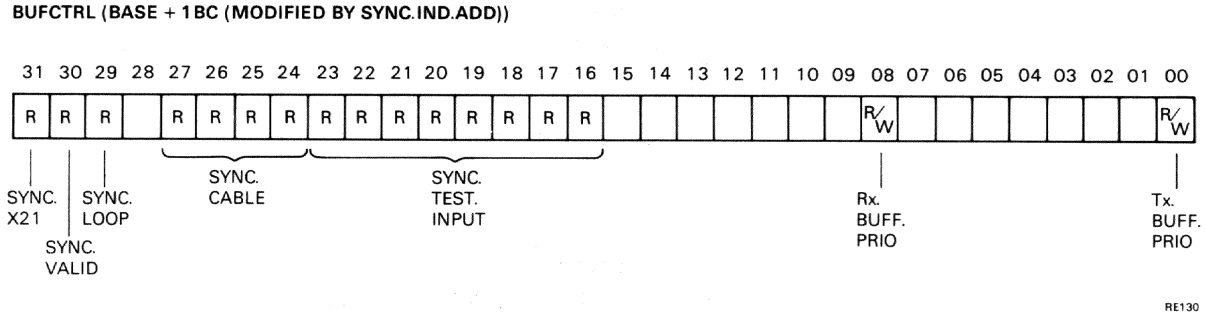
LPR3 (BASE + 1B8 (MODIFIED BY SYNC.IND.ADD))



RE129

Bit	Name	Description
<7:0>	SYNC.CHAR (Sync Character) (R/W)	This character is the sync character used by byte oriented protocols.
<15:8>	RX.MATCH (Receive Match Character) (R/W)	This is the 'match' character used by the GEN BYTE protocol.
<23:16>	ADDRESS1 (First Address Character) (R/W)	The first byte used in address matching.
<31:24>	ADDRESS2 (Second Address Character) (R/W)	The second byte used in address matching.

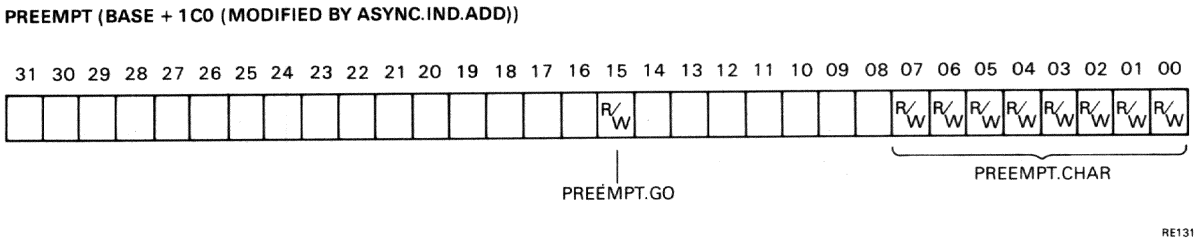
3.3.43 Sync Buffer Control Register (BUFCTRL)



Bit	Name	Description
<0>	TX.BUFF.PRIO (Transmitter Buffer Priority) (R/W)	This bit controls which of the two buffers the device will use if both are valid for DMA when a message is transmitted. If this bit is clear then buffer 1 is used; if it is set then buffer 2 is used. The host should set this bit before it sets up a DMA on buffer 1 and clear it before it sets up a DMA on buffer 2. This makes sure that the 'oldest' buffer is always used when there is a choice.

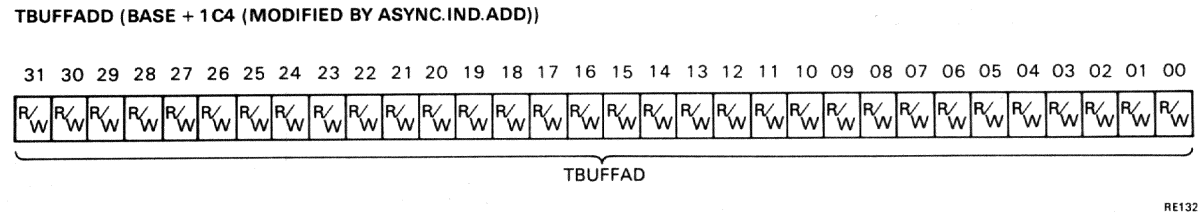
Bit	Name	Description
<8>	RX.BUFF.PRIO (Receiver Buffer Priority) (R/W)	This bit controls which of the two buffers the device will use if both are valid for DMA at the start of a received message. If this bit is clear then buffer 1 is used; if it is set then buffer 2 is used. The host should set this bit before it sets up a DMA on buffer 1 and clear it before it sets up a DMA on buffer 2. This makes sure that the 'oldest' buffer is always used when there is a choice.
<23:16>	SYNC.TEST. INPUT (Test Inputs) (R)	This byte is used during manufacturing testing only.
<27:24>	SYNC.CABLE (Cable Type) (R)	These bits indicate the type of adapter cable or loopback connected at the distribution panel. Bit <27> is the MSB. The meanings of the numerical values are: 0 – no cable present 1 – V.35 cable 2 – V.24 cable 3 – reserved 4 – V.36/RS-422 cable 5 – reserved 6 – reserved 7 – reserved 8 – reserved 9 – unbalanced loopback connector 10 – reserved 11 – reserved 12 – reserved 13 – reserved 14 – balanced loopback connector 15 – reserved
<29>	SYNC.LOOP (Loopback Present) (R)	This bit is used for manufacturing testing only.
<30>	SYNC.VALID (Valid Cable) (R)	This bit indicates that a valid cable is connected to the sync line connector.
<31>	SYNC.X21 (R)	Reserved to DIGITAL.

3.3.44 Async Transmission Preempt Buffer (PREEMPT)



Bit	Name	Description
<7:0>	PREEMPT.CHAR (Character to Transmit) (R/W)	This byte contains the character to be transmitted.
<15>	PREEMPT.GO (Start Preempt) (R/W)	This bit is set by the host when it loads the character to be transmitted. It is cleared by the DMB32 when the character has been read from the register. A TX action is also generated when this bit is cleared. Characters loaded into the register may be transmitted before characters in the transmit FIFO or characters in a current DMA operation.

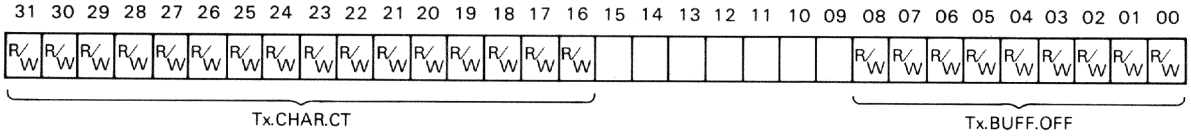
3.3.45 Async Transmit Buffer Address Register (TBUFFAD)



Bit	Name	Description
<31:0>	TBUFFAD (Transmit Buffer Address) (R/W)	Cleared by initialization. This longword is loaded by the host, with the start address of the buffer to be sent by DMA transfer. This register must not be written to between setting TX.DMA.START and the respective TX.ACT being returned. During this period, the contents of this register are undefined. The interpretation of this address is controlled by TX.DMA.PTE AND TX.DMA.PHYS.

3.3.46 Async Transmit Buffer Count/Offset Register (TBUFFCT)

TBUFFCT (BASE + 1C8 (MODIFIED BY ASYNC.IND.ADD))

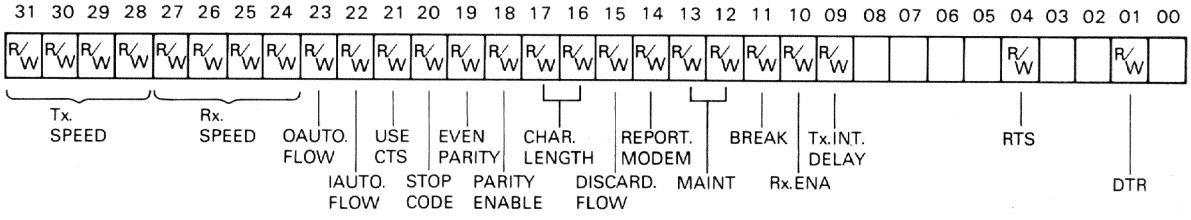


RE133

Bit	Name	Description
<8:0>	TX.BUFF.OFF (Transmit Buffer Offset) (R/W)	These bits contain the offset of the first character of the buffer within the first page containing the buffer. This is only used when TX.DMA.PTE is set.
<31:16>	TX.CHAR.CT (Transmit DMA Character Count) (R/W)	Cleared by initialization. These bits define the length of the Transmit DMA buffer for the selected channel. They are initially loaded by the host and are updated by the DMB32 after a transfer so that they contain the number of characters still to be sent. The number of characters is specified as a 16-bit unsigned integer. This register must not be written to between setting TX.DMA.START and the respective TX.ACT being returned. During this period, the contents of this register are undefined.

3.3.47 Async Line Parameters Register (LPR)

LPR (BASE + 1CC (MODIFIED BY ASYNC.IND.ADD))



RE134

Bit	Name	Description
<1>	DTR (Data Terminal Ready) (R/W)	Cleared by initialization. This bit controls the Data Terminal Ready signal (CCITT Circuit Number 108). When the bit is set, the Data Terminal Ready line goes to the ON state.
<4>	RTS (Request to Send) (R/W)	Cleared by initialization. This bit controls the Request to Send signal (CCITT Circuit Number 105). When the bit is set, the Request to Send line goes to the ON state.
<9>	TX.INT.DELAY (Transmit Interrupt Control) (R/W)	Cleared by initialization. When this bit is clear, the interrupt can take place when the final character has been transferred from host memory. The interrupt indicates that the host can release its buffers and start a new transfer. However, changes made to the line parameter register after this interrupt may affect the last few characters of the buffer. When this bit is set the transmit interrupt for a DMA operation will not take place until the final character has been transmitted. The host can then change the line parameter register (for example, change modem control) knowing that all of the message has been transmitted.
<10>	RX.ENA (Receiver Enable) (R/W)	Cleared by initialization. When this bit is set, the receiver for the selected line is enabled. If RX.ENA is cleared while a character is being assembled, the character is lost.
<11>	BREAK (Break Control) (R/W)	Cleared by initialization. This bit forces the selected line to the Spacing state after the current character has been sent. Transmission continues after the break has been cleared.

Bit	Name	Description
<13:12>	MAINT (Maintenance Mode) (R/W)	Cleared by initialization. These two maintenance bits have the following meaning: 00 Normal Operation 01 Automatic echo mode. Received data is retransmitted (regardless of the state of TX.ENA) at the same speed as the receiver. RX.ENA must be set for this mode to work. Received characters are placed in the receiver FIFO 10 Local loopback. The transmitter output is internally connected to the receiver. Normal received data is ignored and the transmitter data line is held at the Marking state. 11 Remote Loopback. In this mode received data is retransmitted at a clock rate equal to the received clock rate. The data is not placed in the receiver FIFO. The state of TX.ENA is ignored.
<14>	REPORT.MODEM (Report Modem Changes) (R/W)	Cleared by initialization. When this bit is clear, Data Set change information is prevented from being passed to the CPU via the received character FIFO, but the line status register will continue to be updated with the current Data Set status information. This allows these inputs to be used for other control functions. When this bit is set Data Set change information is passed normally.
<15>	DISCARD.FLOW (Discard Flow Control Characters) (R/W)	Cleared by initialization. When this bit is set received flow control characters (XON, XOFF) will not be placed in the received-character FIFO buffer. If OAUTO.FLOW is clear then this bit has no effect.
<17:16>	CHAR.LGTH (Character Length) (R/W)	Set by initialization. These two bits define the character length, excluding the start, stop, and parity bits as follows: 00 – 5 bits per character 01 – 6 bits per character 10 – 7 bits per character 11 – 8 bits per character

Bit	Name	Description
<18>	PARITY.ENAB (Parity Enable) (R/W)	Cleared by initialization. When set, this bit causes a parity bit to be generated and added on transmission, and checked and stripped on reception for the selected line.
<19>	EVEN.PARITY (Even Parity) (R/W)	Cleared by initialization. When PARITY.ENAB is set, this bit controls the sense of the parity according to the following table: 0 – Odd parity 1 – Even Parity
<20>	STOP.CODE (Stop Code) (R/W)	Cleared by initialization. This bit defines the length of the Stop Code at the end of the character. Clear – 1 stop bit for 5-, 6-, 7- or 8-bit characters Set – 2 stop bits for 6-, 7-, or 8-bit characters; 1.5 stop bits for 5-bit characters.
<21>	USE.CTS (CTS Controls Output) (R/W)	Cleared by initialization. When this bit is set, it indicates that a modem should be connected to the port and that the DMB32 should obey the CTS protocol. This is important when the card is sending a DMA buffer without CPU intervention. When the bit is clear, it indicates that a Data-Leads-only type terminal is connected to the port and transmission is to take place irrespective of the state of CTS.

Bit	Name	Description
<22>	IAUTO.FLOW (Automatic Flow Control of Incoming Data) (R/W)	Cleared by initialization. When set, this bit allows the DMB32 to stop internal congestion by requesting lines that are sending characters to stop transmitting. When the Received Character FIFO becomes critically full, the DMB32 will send an XOFF to each line that sends it a character. Thereafter it will send an XOFF to a line in reply to every alternate character received. When the congestion has been removed by the host taking characters from the received character FIFO, an XON will be sent to those lines that have been sent one or more XOFFs. If this bit is clear, NO characters will be sent to this line other than those written to it by the host. Either the DMB32 or the CPU can send flow control characters to any one channel, but not both together. Unpredictable results will occur if the CPU tries to send flow control characters during any time when the DMB32 has been enabled to send flow control characters. However, the CPU can force the DMB32 to send an XOFF or XON by using the SNDOFF bit.
<23>	OAUTO.FLOW (Automatic Flow Control of Outgoing Data) (R/W)	Cleared by initialization. This bit enables the DMB32 to respond automatically to received XON/XOFF characters. When set, this bit causes the DMB32 to update the TX.ENA bit in LNCTRL after receiving flow control characters. A received XOFF causes TX.ENA to be reset and a received XON causes it to be set. The host may also write to TX.ENA while incoming flow control is enabled (for example, to override an XOFF), but care must be taken not to lose any incoming XON characters. The DMB32 may write to the TX.ENA bit for up to 20 microseconds after OAUTO has been cleared by the host. The XONs and XOFFs will still be passed to the host when this feature is enabled. This allows the host to keep track of the line state.

Bit	Name	Description
<27:24>	RX.SPEED (Receive Data Rate) (R/W)	This field is set to 1101 (9600 bits/s) by initialization. These four bits select the async receiver line speed on a line-by-line basis, as specified in the following table.

Value	Data Rate (bit/s)	Max Error (%)	Group
0000	50	0.01	A
0001	75	0.01	B
0010	110	0.08	A and B
0011	134.5	0.07	A and B
0100	150	0.01	B
0101	300	0.01	A and B
0110	600	0.01	A and B
0111	1200	0.01	A and B
1000	1800	0.01	B
1001	2000	0.19	B
1010	2400	0.01	A and B
1011	4800	0.01	A and B
1100	7200	0.01	A
1101	9600	0.01	A and B
1110	19200	0.01	B
1111	38400	0.01	A

<31:28>	TX.SPEED (Transmit Data Rate) (R/W)	This field is set to 1101 (9600 bits/s) by initialization. These four bits select the async transmitter line speed according to the same table as given for RX.SPEED. When selecting split-speed operation the transmit and receive speeds of both channels sharing a DUART must be contained in the same group (A or B). If a channels transmitter and receiver are configured in conflicting groups, the group of the receiver will be used; if two channels sharing a DUART are configured in conflicting groups, the group of the most recently configured channel will be used.

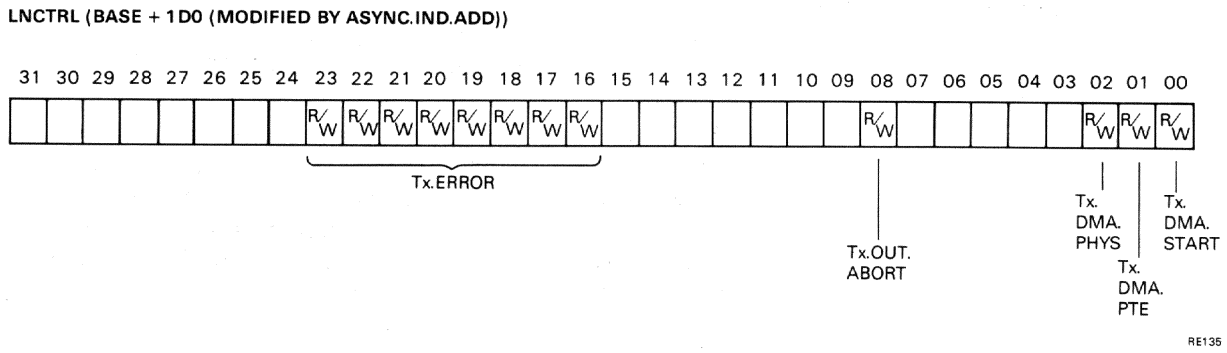
Bit	Name	Description
		If the speed group of a port is changed implicitly (that is, by configuring a partner port in a conflicting group), the new speed of that port is given by:

Previously Selected Speed (bits/s)	New Speed (bits/s)
50	75
75	50
150	200
1800	7200
2000	1050
7200	1800
19200	38400
38400	19200

The ports are implemented using Dual UARTs and are associated in the following manner:

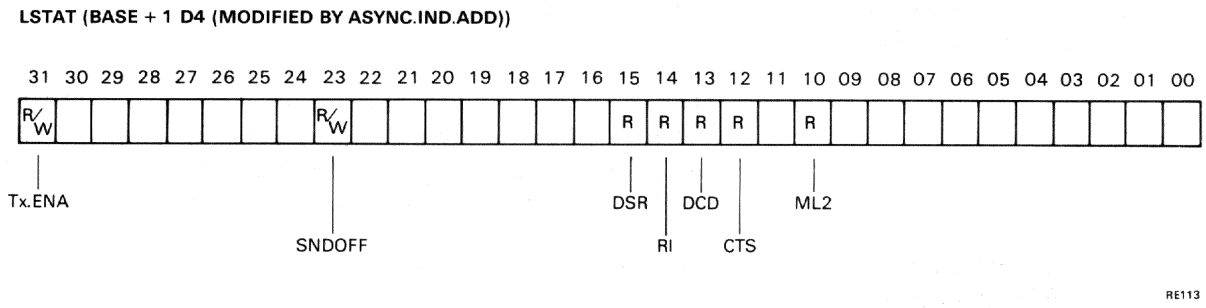
Ports 0 and 1 use the same DUART
Ports 2 and 3 use the same DUART
Ports 4 and 5 use the same DUART
Ports 6 and 7 use the same DUART

3.3.48 Async Line Control Register (LNCTRL)



Bit	Name	Description
<0>	TX.DMA.START (Start DMA Transfer) (R/W)	Cleared by initialization. This bit is set by the host to start an async DMA transfer. Having set this bit, the host must not write to TBUFFCT or TBUFFAD, or the byte containing this bit until the TX.ACT has been received to say that this transfer has finished. This bit will be reset before the TX.ACT is returned.
<1>	TX.DMA.PTE (PTE Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is the address of the first PTE for the buffer. If this bit is clear, then the DMB32 assumes that the address is the address of the first byte of the buffer. If this bit is set, the DMB32 assumes that the address is the address of a PTE which describes the page which contains the first byte of the buffer. This address must be long-word aligned and the offset of the first byte is contained in TX.BUFF.OFF. This bit may not be modified by the host while a DMA is in progress.
<2>	TX.DMA.PHYS (Physical Address) (R/W)	Cleared by initialization. This bit is set by the host to indicate that the DMA address is a physical address. If this bit is clear then the DMB32 assumes that the address is a system virtual address. This bit may not be modified by the host while a DMA is in progress.
<8>	TX.OUT.ABORT (Transmitter Output Abort) (R/W)	Cleared by initialization. This bit is set by the CPU to abort transmission of data. On seeing this bit set, the DMB32 stops transmitting data on this channel. If a DMA was in progress it updates the DMA address/count register. After a DMA has been aborted, the parameters in the DMA address and count registers will be in a suitable form to continue the transfer by clearing the TX.DMA.ABORT bit and setting the TX.DMA.START bit.
<23:16>	TX.ERROR (Transmitter Error Bits) (R/W)	Cleared by initialization. If this byte contains any value other than zero, this indicates that errors have have occurred in the transmitter during the DMA operation. Error codes are described in Section 3.4.8.

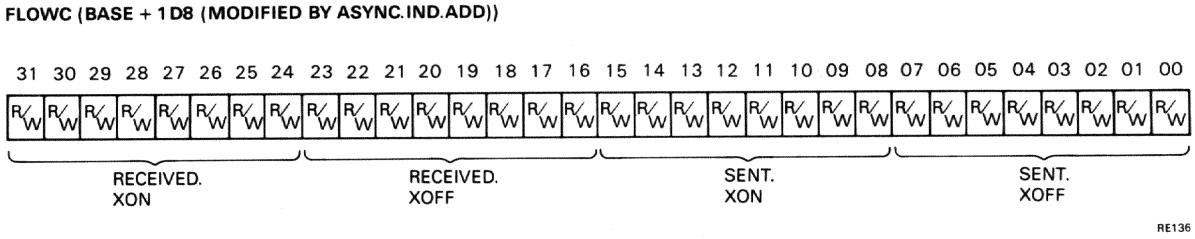
3.3.49 Async Line Status Register (LSTAT)



Bit	Name	Description
<10>	ML2 (Spare Modem Control Lead) (R)	Indicates the current status of the second modem loopback signal (Test Indicator) from the modem.
<12>	CTS (Clear to Send) (R)	Indicates the current status of the CTS signal from the modem. The host should not write to the byte containing this and the other modem control bits. If it does, then it may read incorrect modem status until the correct status is made available.
<13>	DCD (Data Carrier Detected) (R)	Indicates the current status of the DCD signal from the modem.
<14>	RI (Ring Indicator) (R)	Indicates the current status of the Ring Indicator signal from the modem. The Ring Indicator signal is monitored over a period of 30 ms, and the state is not updated until all the samples taken over this period are in agreement.
<15>	DSR (Data Set Ready) (R)	Indicates the current status of the DSR signal from the modem.
<23>	SNDOFF (Send XOFF) (R/W)	Cleared by initialization. When this bit is set, the DMB32 will behave as if its internal FIFO alarm condition has been reached. That is, it will transmit XOFF characters to every other received character. It will transmit an XON character when this bit is cleared if it has sent an XOFF character. It does this regardless of the state of IAUTO.

Bit	Name	Description
<31>	TX.ENA (Transmitter Enable) (R/W)	Set by initialization. When set, the DMB32 will transmit characters as requested. When clear, the DMB32 will only transmit internally generated flow control characters and preempt characters. This bit will be updated by the DMB32 on receiving of flow control characters if the OAUTO bit in LPR is set.

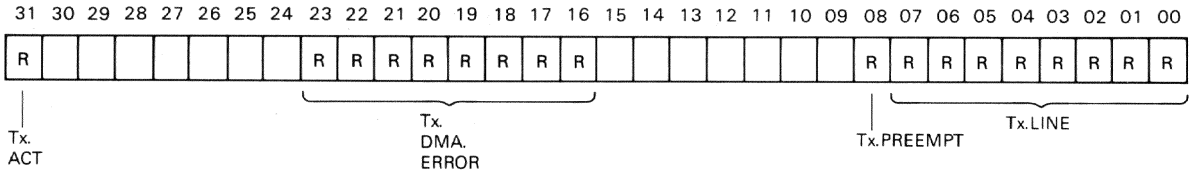
3.3.50 Async Flow Control Characters (FLOWC)



Bit	Name	Description
<7:0>	SENT.XOFF (Transmitted XOFF) (R/W)	This field is set to CTRL/S by Master Reset. This byte contains the character that the DMB32 will use when it sends an XOFF.
<15:8>	SENT.XON (Transmitted XON) (R/W)	This field is set to CTRL/Q by Master Reset. This byte contains the character that the DMB32 will use when it sends an XON.
<23:16>	RECEIVED.XOFF (Received XOFF) (R/W)	This field is set to CTRL/S by Master Reset. This byte contains the character that the DMB32 will treat as an XOFF when it is received.
<31:24>	RECEIVED.XON (Received XON) (R/W)	This field is set to CTRL/Q by Master Reset. This byte contains the character that the DMB32 will treat as an XON when it is received.

3.3.51 Async Transmit Completion FIFO (TBUF)

TBUF (BASE + 204)



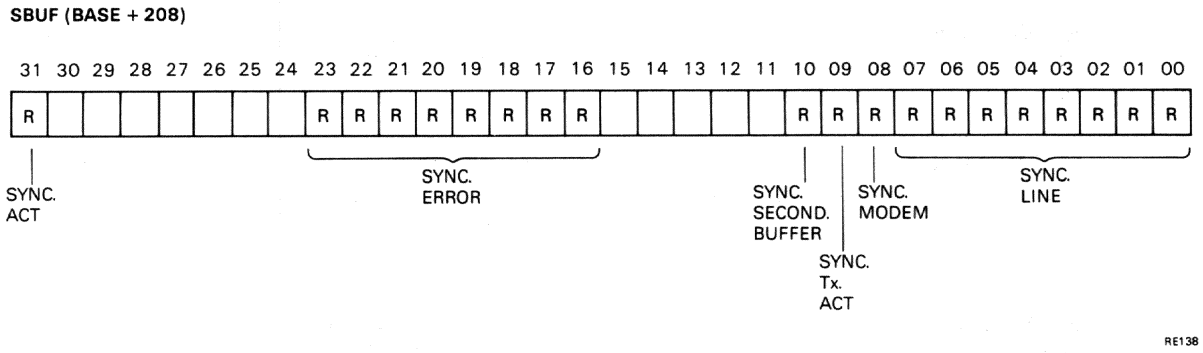
NOTE

The host must not write to this FIFO when TX.ACT (bit <31> is clear. If TX.ACT is set, the host must acknowledge a read from this FIFO by writing to it, which will clear TX.ACT.

Bit	Name	Description
<7:0>	TX.LINE (Transmit Line Number) (R)	Undefined if TX.ACT is clear. If TX.ACT is set then this byte contains the number of the line on which one of the following has just occurred: • The preempt character register has been emptied. • A DMA transfer has completed normally. • A DMA abort sequence has been completed • An error has been detected during a DMA transfer.
<8>	TX.PREEMPT (Preempt Completed) (R)	This bit indicates that the action is because a single character (preempt) output has completed. It is clear if a DMA has completed.
<23:16>	TX.DMA.ERROR (Transmit Error Code) (R)	This byte will contain a non-zero error code if an error is detected while performing output. The same value is in TX.ERROR

Bit	Name	Description
<31>	TX.ACT (Transmitter Action) (R)	Cleared by initialization. This bit is set by DMB32 to report one of the following events concerning the channel specified in TX.LINE: • The preempt character register has been emptied. • A DMA buffer previously passed to the DMB32 has been completed (last character transmitted). • A DMA abort sequence has been completed. • Transmission of a DMA buffer has been terminated by DMB32 because a nonexistent memory location was specified, or the read data had a parity error. The host must not write to this FIFO when this bit is clear. If this bit is set, the host must acknowledge a read from this FIFO by writing to it to clear this bit.

3.3.52 Sync Line Completion FIFO (SBUF)

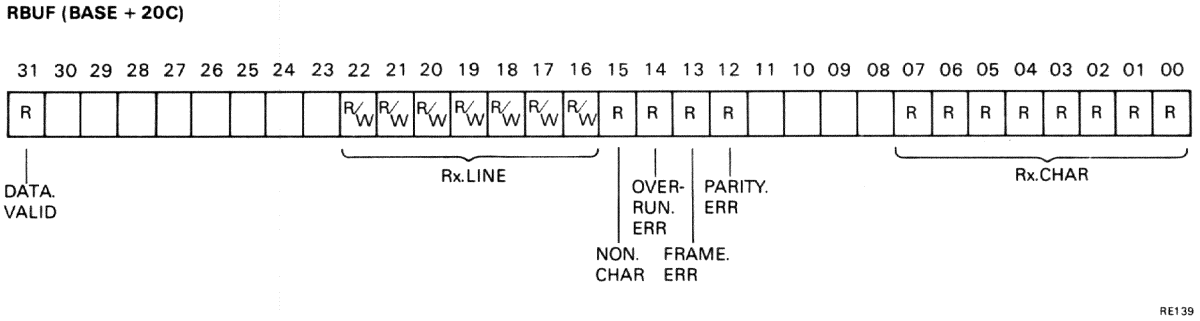


NOTE

The host must not write to this FIFO when SYNC.ACT (bit <31> is clear. If SYNC.ACT is set, the host must acknowledge a read from this FIFO by writing to it, which will clear SYNC.ACT.

Bit	Name	Description
<7:0>	SYNC.LINE (Sync Line Number) (R)	Undefined if SYNC.ACT is clear. If SYNC.ACT is set then this byte contains the number of the line on which one of the following has just occurred: • A DMA transfer has completed normally, or with an error • A modem change has been detected.
<8>	SYNC.MODEM (Modem Change) (R)	This bit is set if the FIFO entry is due to a modem change.
<9>	SYNC.TX.ACT (Sync Transmit Complete) (R)	This bit is set if the FIFO entry is due to completion of a transmission. If this bit is clear and SYNC.MODEM is clear then the FIFO entry is due to completion of a receive operation.
<10>	SYNC.SECOND.BUFFER (Buffer Number) (R)	This bit is set if the FIFO entry is due to completion of a DMA operation on the second receive or transmit buffer.
<23:16>	SYNC.ERROR (Sync Line Error Code) (R)	This byte contains the error code for the operation (see TX1.ERROR, RX1.ERROR).
or	SYNC.MODEM (Sync Line New Modem Status) (R)	When the FIFO entry relates to a modem status change operation, this byte contains the new modem status rather than an error code. This status is of the same format as LPR2<15:8>.
<31>	SYNC.ACT (Sync Action) (R)	Cleared by initialization or when the sync Action FIFO is empty. This bit is set by DMB32 to report one of the following events concerning the channel specified by sync line number: • A DMA operation has completed • A modem status change has been detected The host must not write to this FIFO when this bit is clear. If this bit is set, the host must acknowledge a read from this FIFO by writing to it to clear this bit.

3.3.53 Async Receive Buffer (RBUF)



NOTE

The host must not write to this FIFO when DATA.VALID (bit <31> is clear. If DATA.VALID is set, the host must acknowledge a read from this FIFO by writing to it, which will clear DATA.VALID.

Bit	Name	Description
<7:0>	RXCHAR (Received Character) (R)	If the NON.CHAR bit (RBUF<15>) is clear, these bits contain the oldest received character on the line defined by RX.LINE<7:0>. Characters are received least significant bit first and this bit is placed in RBUF<0>. Characters of less than 8 bits are right justified with the high-order bits set to zero. DMB32 does not contain a break-detect bit. Line breaks are passed to the CPU as a single null character with the framing-error bit (RBUF<13>) set. If the NON.CHAR bit is set, these bits contain either the current Data Set status following a recent change in status (RBUF<0> = 0), or a diagnostic code (RBUF<0> = 1). RBUF<0> = 0 (indicates Data Set change) RBUF<1> = Undefined RBUF<2> = Spare modem control input (Test Indicator) RBUF<3> = Undefined RBUF<4> = Clear To Send RBUF<5> = Data Carrier Detected RBUF<6> = Ring Indicator (integrated over 30 ms) RBUF<7> = Data Set Ready The meaning of the bits in RBUF<7:1> during a diagnostic report is described in Chapter 4, Section 4.3.2.1. After initialization of the DMB32 there will be diagnostic reports in the received character FIFO if it has failed self-test.

Bit	Name	Description
<12>	PARITY.ERR (Parity Error) (R)	This bit is set if parity was enabled for the line on which a character was received with incorrect parity.
<13>	FRAME.ERR (Framing Error) (R)	This bit is set if the line on which a character was received was in the spacing state at the time the first stop bit was sampled.
<14>	OVERRUN.ERR (Overrun Error) (R)	This bit is set if one or more characters on the line are lost because the FIFO is full or the DMB32 fails to service the DUARTs. In the event of an overrun, the DUART's three-character FIFO is cleared, and a null character with an overrun error is placed in the received character FIFO. This is a dummy character and was not in the original character stream. Subsequent characters will be as read from the DUART.
<15>	NON.CHAR (Non Character Data) (R)	This bit is set when the data read from the FIFO is not a character. The data may be either a modem change indication or a diagnostic message.
<23:16>	RX.LINE (Receive Line Number) (R/W)	This byte contains the line number on which data has been received, or a Data Set change or a diagnostic report has occurred.
<31>	DATA.VALID (Data Valid) (R)	This bit is cleared by master reset or by the FIFO becoming empty. It is set when the first character is loaded into the FIFO, and remains set as long as there is valid data in the FIFO. The host must not write to this FIFO when this bit is clear. If this bit is set, the host must acknowledge a read from this FIFO by writing to it to clear this bit.

3.4 PROGRAMMING FEATURES

3.4.1 Initialization

The DMB32 is initialized by its on-board firmware. Initialization takes place after a bus reset sequence, or when the host sets MAINT<1> (PROG.RESET).

Before starting initialization, the on-board diagnostics run a self-test program. The results of this test are reported by diagnostic codes placed in the Received Character FIFO (RBUF) and the General Purpose Register 0 (GPR0) which, in the DMB32, is also designated the Test Summary Register (TSMR).

NOTE

This self-test diagnostic can be skipped on command from the program. This is covered in Chapter 4, Section 4.3.

The state of the DMB32 after a successful self-test is as follows:

1. The self-test status (STS) bit (BICSR<11>) is set.

2. The BROKE bit (BICSR<12>) is cleared.

3. All async channels are set for:

   a. Send and receive 9600 bits/s
   b. Eight data bits
   c. One stop bit
   d. One start bit (cannot be altered)
   e. Parity disabled
   f. Parity odd
   g. Auto-flow off
   h. RX disabled
   i. TX enabled
   j. No break on line
   k. No UART loopback
   l. Data-leads-only circuit
   m. DTR and RTS off
   n. TX.DMA.START cleared
   o. TX.DMA.ABORT cleared.

4. All sync channel fields (except BUFCTRL<31:16>) set to 0.

5. All printer port fields set to 0.

The DMB32 clears the PROG.RESET bit (MAINT<1>) and the INIT bit (BICSR<13>) when initialization and self-test are complete.

3.4.2 Configuration

After the DMB32 self-initialization, the driver program can configure the DMB32 as needed. This is done using the line parameter registers.

By writing to the associated LPR the program can select data rate, character length, parity and stop bit length for each async channel. Individual receivers and transmitters can be enabled and auto-flow selected.

By writing to LPR1, LPR2, and LPR3 the program can select the protocol, mode, and individual characteristics for the sync channel.

There is no equivalent of LPR for the printer port, although there are the registers PFIX, PCTRL and PCAR. Control and formatting commands are embedded in the data stream by host software.

3.4.3 Using the FIFOs

The contents of TBUF, SBUF, and RBUF are not cleared by a READ action from the FIFO register. Therefore, after each READ the host should write to the FIFO (the data associated with the WRITE action is not used). This action will remove the top entry of the FIFO.

The data in the FIFOs will only be valid if bit <31> is set. The host must not write to any of these FIFOs if bit <31> is clear.

3.4.4 Receiving via the RX FIFO

Characters received on the async channels are merged with the channel number and any error status, and are put in the received character FIFO (RBUF). The FIFO can be read by any processor connected on the VAXBI.

If a character is put into an empty RBUF, the DMB32 sets DATA.VALID. This bit will remain set as long as there is valid data in RBUF. If RXIE is also set, the host will be interrupted at the async vector. The host's interrupt routine should then remove data from RBUF until DATA.VALID is clear.

If RXIE is not set, the host must poll RBUF often enough to prevent data loss.

3.4.5 Preempt Transfers

Async channels can transmit single characters by using the Preempt register. The host has only to make sure that the correct line number is loaded in ASYNC.IND.ADD and then write the character and the PREEMPT.GO bit to the preempt register. The character will be transmitted before any outstanding DMA transfers are serviced on that channel.

After the DMB32 transfers the character from PREEMPT, it writes the channel number, and sets TX.ACT and TX.PREEMPT in TBUF. This adds the TX action report to any queue in TBUF. If TXIE is set, and the TX interrupt is not already raised to report previous entries, the host will be interrupted at the transmit vector. Further interrupts will not be produced until TBUF is cleared.

3.4.6 DMA Operations

DMA access to host memory is essentially the same for sync, async, and printer channels (however, only the sync channel uses DMA transfer on received data).

The DMB32 always attempts to transfer DMA buffers an octaword at a time. Figure 3-2 shows how the DMB32 handles buffers that are not octaword aligned.

During a DMA WRITE to host memory, that part of the octaword that is outside the buffer will be masked out by a Write Mask transaction.

During a DMA READ from host memory, the full octaword at each end of the buffer is read by the DMB32, but that part of the octaword that is outside the buffer will be discarded.

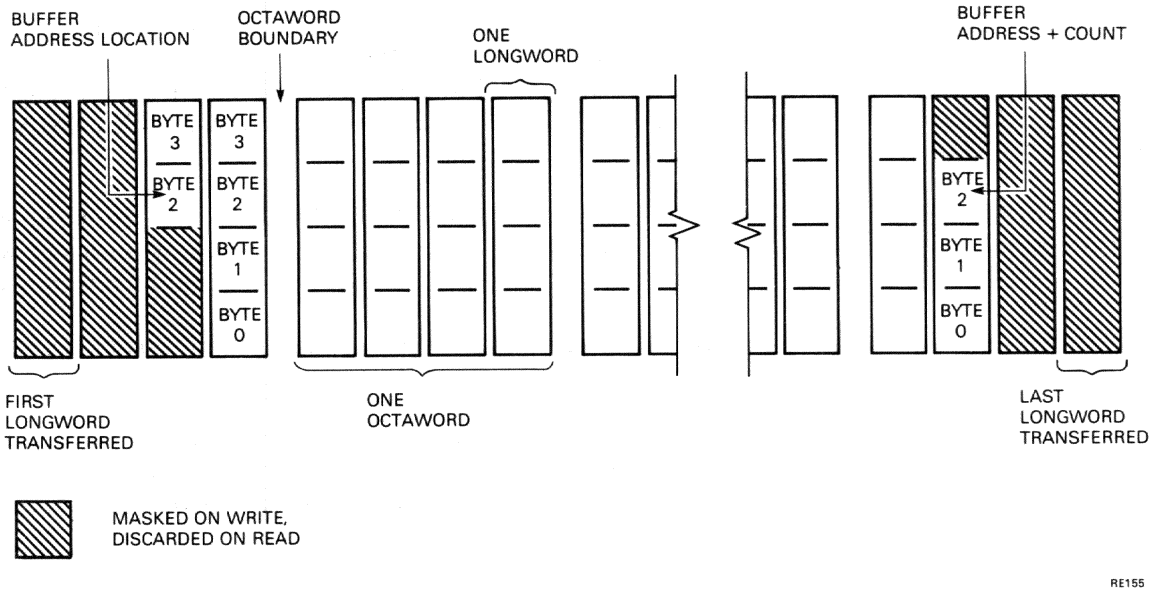


Figure 3-2 Handling an Unaligned DMA Buffer

3.4.6.1 Address Translation – DMA buffer information supplied by the host must define the location of the buffer, and its size. The buffer can be defined in one of four ways, in accordance with VAX architecture.

1. Physical address and size (no translation)
2. Physical address of the first PTE for the buffer, the offset within the first page, and the size.
3. System virtual address and size
4. System virtual address of the first Process PTE for the buffer, the offset within the first page, and the size.

When TXn.DMA.PHYS is 0 the address is virtual and the DMB32 uses the contents of the SPTE, SPTS, GPTE and GPTS registers to access the appropriate page tables in host memory. Using these, and any offsets supplied by the host, the option constructs the physical addresses of the pages. Address translation of each page is performed separately and continues during the transfer. Accesses to page tables are done by longword DMA transfers.

3.4.6.2 Transmitting Data – In this description the registers for the sync channel buffer number one will be used as examples. The async and printer channels have register names that are recognizable as equivalents. For example, TBUFFADD and PBUFFAD are equivalent to TBUFFAD1.

The host should make sure that TX1.DMA.START is clear before setting up the transfer of a DMA buffer. If not, there is a DMA transfer in progress. The start address and size of the buffer, and the address offset if relevant, can then be written to TBUFFAD1 and TBUFFCT1. The transfer is initiated by TX1.DMA.START, so it must always be written last. Therefore it is logical to write TX1.DMA.PHYS, TX1.DMA.PTE and TX1.DMA.START at the same time.

The contents of TBUFFCT1 and TBUFFAD1 are invalid while TX1.DMA.START is set. These will not be updated until the transfer is completed, stopped by an error or aborted. The host must not write to TBUFFCT1 or TBUFFAD1 until the DMB32 clears TX1.DMA.START at the end of transmission.

After TX1.DMA.START is set, the DMB32 will perform the transaction and will report via SBUF. If SYNC.I.E is set, the host will be interrupted at the sync vector.

To abort a DMA transmission, the host must set TX1.DMA.ABORT. The DMB32 will then stop transmission and update TBUFFCT1 and TBUFFAD1 to reflect the number of characters still to be transmitted. TX1.DMA.START will be cleared and a sync action report placed in SBUF. The host will be interrupted if SYNC.I.E is set.

3.4.6.3 Receiving Data – Reception of a DMA buffer is much the same as transmission. In this case however, the relevant registers and fields are:

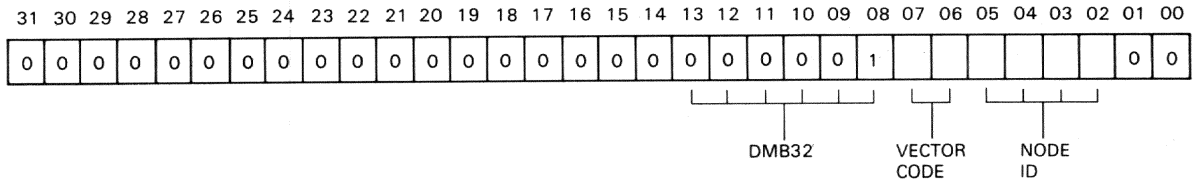
Register	Field(s)
RBUFFAD1	RBUFFAD1
RBUFFCT1	RX.BUFF.CT1 RX.BUFF.OFF1
RLNCTRL1	RX1.DMA.PHYS RX1.DMA.PTE RX1.DMA.START RX1.DMA.ABORT

When the DMA has been completed, stopped by an error or aborted, the DMB32 will place a sync action report in SBUF. If SYNC.I.E is set, the host will be interrupted at the sync vector.

3.4.7 Interrupt Control

In addition to the bus error interrupt vector generated by the BIIC, the DMB32 can provide three interrupt vectors. These vectors are constructed as shown by Figure 3-3. Bits <31:9> are always zero, bit <8> indicates that the interrupts are type 1 VAXBI interrupts.

The three DMB32 vectors report several different events, therefore host interrupt routines must check the DMB32 registers to find the precise reason for the interrupt. (See Error Codes, Section 3.4.8)



VECTOR CODE			INTERRUPT
BITS 7, 6			
0	D	BIIC ERROR	PRINTER CHANNEL
D	1	SYNC CHANNEL	ASYNC CHANNELS
1	0	ASYNC CHANNEL	SYNC CHANNEL
1	1	PRINTER CHANNEL	

RE140

Figure 3-3 DMB32 Interrupt Vectors

Interrupts are raised to report the following conditions.

Async channels
- A character has just been transferred from a preempt register
- An async DMA transfer has been stopped by an error, aborted or completed with or without an error
- A character has just been placed in an empty RX FIFO The interrupt may be delayed by RX.TIMER. If REPORT.MODEM is set, the delay will be overridden by any change of modem status on an async channel.
- The interrupt is not already raised, and async modem change information is placed in the RX FIFO

Sync channel
- A sync DMA transfer has been stopped by an error, aborted or has completed with or without error
- Sync modem change information has been placed in SBUF

Printer channel
- A printer DMA transfer has been stopped by an error, aborted or has completed with or without error

Interrupts are enabled/disabled by RX.I.E, TX.I.E, SYNC.I.E and PR.I.E bits in the ACSR, ACSR and PCSR.

3.4.8 Error Codes

When an interrupt condition occurs, the DMB32 provides status and error information in several registers.

- RBUF, TBUF and LNCTRL hold async status and errors
- SBUF, TLNCTRLn and RLNCTRLn hold sync status and errors.
- PCTRL and PCSR hold printer status and errors

The error codes in bits <23:16> of TBUF, LNCTRL, SBUF, TLNCTRLn, RLNCTRLn and PCTRL are described in this section. Other status and error codes are described in the appropriate register bit descriptions of Section 3.3, or in the Chapter 4, Section 4.3.2.1 (Self-Test error codes in the RX FIFO).

The format of bits<23:16> in all the above registers is the same. The top four bits indicate the error; the low four bits supply further information about the error. Bits<23:20> are interpreted as follows:

Bits	23	22	21	20	Hexadecimal Value	Error
	0	0	0	0	0	No error
	0	0	0	1	1	DMA error
	0	0	1	0	2	Message error
	0	0	1	1	3	Last character incomplete
	0	1	0	0	4	Buffer error
	0	1	0	1	5	Modem error
	0	1	1	0	6	Aborted by host
	1	0	0	0	8	Printer not connected
	1	0	0	1	9	Internal error

The interpretation of bits<19:16> depends upon the error. This is explained in the following sections.

3.4.8.1 No Error – In this case bits<19:16> are always 0.

3.4.8.2 DMA Error – This error occurs when the DMB32 was not able to perform a DMA operation to the specified buffer (or to the page tables describing the buffer).

After a DMA has completed, the byte count register will indicate the number of characters that were not transferred to/from host memory. The address, although valid, may have been swapped to a disk and will not therefore be in the buffer following text). For a DMA error, the code in bits<19:16> can be interpreted as follows.

Bits	17	16	Error relates to:
	0	0	A system PTE
	0	1	A global PTE
	1	0	A process PTE
	1	1	The buffer itself

Bits	19	18	
	0	0	Memory did not respond. In this case the buffer address register holds the physical address of the failing memory location. (The address may be rounded down to the octaword boundary below the actual failing address).
	0	1	Uncorrectable error in the memory (for example, parity). In this case the buffer address register holds the physical address of the failing memory location. (The address may be rounded down to the octaword boundary below the actual failing address).
	1	0	PTE was not valid. In this case the buffer address register holds the address (either physical or virtual) of the bad PTE.
	1	1	PTE did not allow the required access (for example, kernel mode READ or WRITE). In this case the buffer address register holds the address (either physical or virtual) of the bad PTE.

3.4.8.3 Message Error – This error occurs when a received message is invalid. Bits<19:16> indicate the error that was detected.

Bits	19	18	17	16	Hexadecimal Value	Error
	0	0	0	1	1	Bad block-check in header (DDCMP only)
	0	0	1	0	2	Bad block-check in data
	0	0	1	1	3	Message longer than buffer with good block check
	0	1	0	0	4	Message longer than buffer with bad VRC
	0	1	0	1	5	Abort character received
	0	1	1	0	6	Invalid character received

3.4.8.4 Last Character Incomplete – This error only occurs for received bit-oriented protocol messages. Bits<19:16> indicate the number of bits that were missing from the last character.

3.4.8.5 Buffer Error – Indicates an error in the buffer to be transmitted.

Bits	19	18	17	16	Hexadecimal Value	Error
	0	0	0	1	1	Invalid header
	0	0	1	0	2	Invalid message length in header
	0	0	1	1	3	Invalid character in the message

3.4.8.6 Modem Error – Indicates a problem with a modem or its cable.

Bits	19	18	17	16	Hexadecimal Value	Error
	0	0	0	1	1	Modem clock stopped (No modem clock was detected for at least one second)
	0	0	1	0	2	Invalid modem status
	0	1	0	0	4	Cable error (The code signals from the cable have changed, or no valid cable is present)

3.4.8.7 Aborted by Host – This indication occurs when the host aborts an operation.

Bits	19	18	17	16	Hexadecimal Value	Error
	0	0	0	1	1	Aborted by the host

3.4.8.8 Printer Offline – Indicates that the printer Connect Verify signal is not asserted.

Bits	19	18	17	16	Hexadecimal Value	Error
	0	0	0	1	1	The printer is not connected.

3.4.8.9 Internal Error – Indicates that the DMB32 detected an internal error.

Bits	19	18	17	16	Hexadecimal Value	Error
	0	0	0	1	1	Data overrun – the host memory was not able to handle all the data received from the communications channels. (This may be the result of attempting to run a sync protocol faster than it is able to go.)
	0	0	1	0	2	Data underrun – the DMB32 was not able to move data fast enough from host memory. (This may be the result of attempting to run a sync protocol faster than it is able to go.)

3.4.9 Automatic Flow-Control

Flow-control is the control of data flow on a communications channel. Its purpose is to prevent loss of data due to overrun of the FIFOs or communications channels. Control is achieved by embedding flow control characters XON and XOFF in the data stream.

A channel that receives an XOFF stops sending characters until it receives an XON. A channel that is in danger of being overrun by received data, sends an XOFF. It sends an XON when the congestion is relieved.

In a system that includes a DMB32, the host can choose from three methods of flow-control.

1. Use driver software to issue and respond to XONs and XOFFs
2. Program the DMB32 to do this automatically (auto-flow)
3. Directly command the DMB32 when to issue XONs and XOFFs

Method 1 causes the greatest software overhead, and creates delay in responding to received XOFFs. It does not stop the DMB32 from being overrun but it does allow the host to prevent overflow of its internal buffers.

Method 2 relieves the host software overhead and improves the response to received XOFFs. The DMB32 will not overrun but host buffers may.

Method 3 is applied to received data only. It causes some software overhead, and will protect host buffers but not the DMB32.

Methods 2 and 3 can be combined to prevent data loss at the host, the DMB32, or the equipment at the other end of the communications channel.

If both the host and the DMB32 perform flow-control on the same channel, it is difficult to keep track of XONs and XOFFs that have been sent and received. Therefore method 1 should not be combined with any other method.

NOTE

XONs and XOFFs issued by the DMB32 will be transmitted even if TX.ENA is cleared. XONs and XOFFs inserted in the data stream by the host will not.

The DMB32 can be programmed for automatic flow control (auto-flow) on any or all async channels. If so programmed, it will automatically regulate the flow of characters. Four bits control this function.

IAUTO.FLOW - LPR<22>
SNDOFF - LSTAT<23>
OAUTO.FLOW - LPR<23>
DISCARD.FLOW - LPR<15>

IAUTO.FLOW and SNDOFF both operate on received characters. IAUTO.FLOW is an enable bit that allows the state of the RX FIFO to control the generation of XON and XOFF codes. SNDOFF is a direct command from the host. It can be used to prevent overflow of data buffers in the host. If SNDOFF is set, the DMB32 will behave as though the RX FIFO is critical (see following text).

OAUTO.FLOW and DISCARD.FLOW operate on transmitted characters. OAUTO.FLOW is an enable bit that allows the DMB32 to respond to XONs and XOFFs from the channel. DISCARD.FLOW controls the reporting of such XONs and XOFFs to the host.

3.4.9.1 IAUTO.FLOW - The DMB32 hardware recognizes when the RX FIFO:

a. Is full

b. Was 3/4 full and is not yet down to 1/2 full (critical)

c. Is empty

d. Has just become 'not empty'

The firmware uses state b for auto-flow control.

If the host sets a channel's IAUTO.FLOW bit, the DMB32 will send that channel an XOFF if it receives a character after the RX FIFO becomes 3/4 full. If the channel does not respond to XOFF, the DMB32 will send an XOFF in reply to every alternate character received on that channel. An XON will be sent when the RX FIFO becomes less than half full unless SNDOFF for that channel is set. XONs are only sent to channels to which an XOFF has been sent.

3.4.9.2 SNDOFF – When SNDOFF is set, the DMB32 sends an XOFF and then acts as if IAUTO.FLOW is set and the RX FIFO is critical. When SNDOFF is reset, an XON will be sent unless IAUTO.FLOW is set and the RX FIFO is critical.

NOTE
Whenever both SNDOFF and IAUTO.FLOW become clear while the channel is in the XOFF state, an XON is sent immediately.

3.4.9.3 OAUTO.FLOW – If the host sets OAUTO.FLOW, the DMB32 will automatically respond to XON and XOFF characters from the channel. It does this by setting and clearing the TX.ENA bit. Up to three characters may be transmitted after XOFF has been received.

The host program may also control the TX.ENA bit. In this case it is important to keep track of received XON and XOFF characters.

NOTE
The DMB32 may change the state of TX.ENA for up to 20 microseconds after OAUTO.FLOW is cleared by the host.

3.4.9.4 DISCARD.FLOW – When OAUTO.FLOW is set, received XON and XOFF characters will be reported via RBUF unless DISCARD.FLOW is set. When DISCARD.FLOW is set, any received XONs and XOFFs will be discarded. DISCARD.FLOW has no effect if OAUTO.FLOW is clear; received XONs and XOFFs will be treated as data.

3.4.9.5 Flow-Control State Diagrams – State diagrams for DMB32 flow-control operations follow. Figures 3-4 and 3-5 are for auto-flow of transmitted and received data. Figure 3-6 is for program initiated flow-control by the SNDOFF bit.

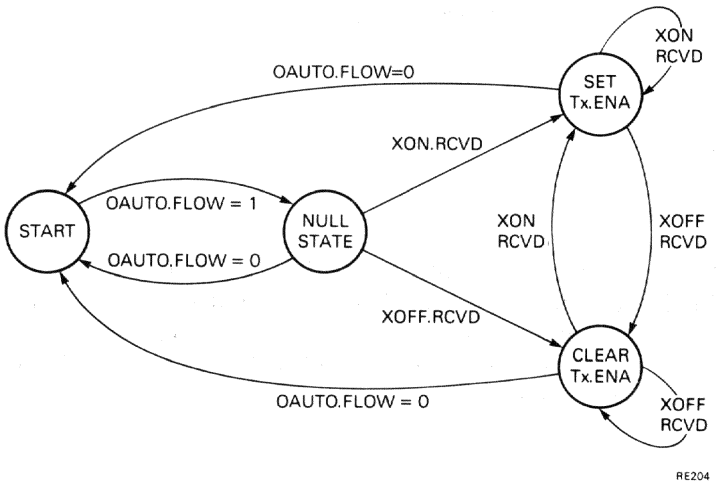


Figure 3-4 Automatic Flow-Control of Transmitted Characters

NOTE
The RX FIFO becomes critical when it is 3/4 full. It remains critical until it is less than 1/2 full.

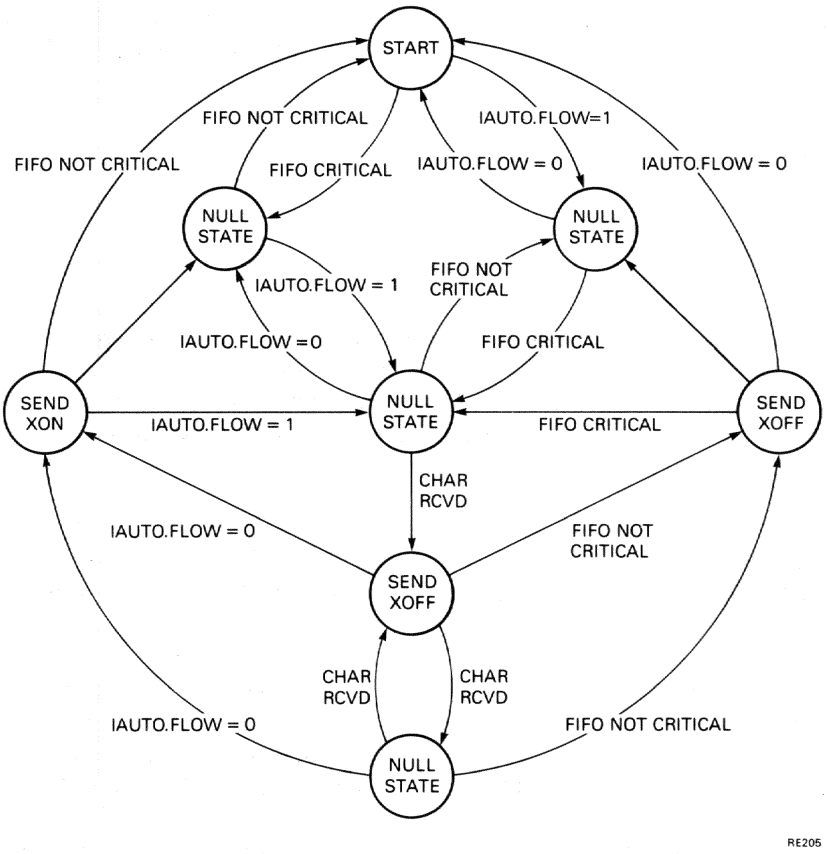


Figure 3-5 Automatic Flow-Control of Received Characters

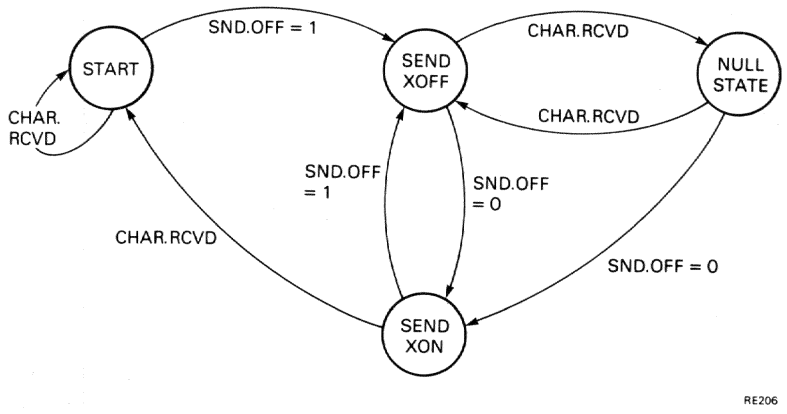


Figure 3-6 Program Initiated Flow-Control

3.4.9.6 Flow-Control Characters – The host can select the codes to be used as received and transmitted XONs and XOFFs on a channel by writing to the channel's FLOWC register. The default values for all channels are:

Received XOFF	=	CTRL/S	=	13_{16}
Transmitted XOFF	=	CTRL/S	=	13_{16}
Received XON	=	CTRL/Q	=	11_{16}
Transmitted XON	=	CTRL/Q	=	11_{16}

NOTE

When checking for flow-control characters, the DMB32 only checks characters that do not have transmission errors. The parity bit is stripped and the remaining bits are checked for XON and XOFF codes.

3.4.10 Async Modem Control

Each async channel provides modem control outputs DTR and RTS, and monitors modem status inputs ML2, CTS, DSR, DCD and RI. These signals can be used for modem control or as general purpose inputs and outputs.

CTS, DSR and DCD are sampled every 10 ms. Therefore, to make sure a change is detected, these signals must remain steady for at least 10 ms after a change of state. RI is also sampled every 10 ms, but it must remain steady for three consecutive samples (at least 30 ms) before a change of state is recognized. A copy of CTS, DSR, DCD and RI is maintained in LSTAT. It is updated after any change.

The only hardware control between the modem control logic and the receiver and transmitter logic is USE.CTS (see below). Therefore any coordination should be done under program control. Two control bits, REPORT.MODEM and USE.CTS, are provided to support such coordination.

If REPORT.MODEM is set, modem status changes will not only be recorded in LSTAT but will also be reported via the RX FIFO. Thus the host can use the appropriate protocol to control the channel. If REPORT.MODEM is clear, reports via the FIFO are inhibited. By polling LSTAT, control and status can still be managed by the host if required.

When the DMB32 is transmitting a DMA buffer, the host has no control over CTS protocol. By setting USE.CTS, the host can stop the DMB32 from transmitting until CTS is asserted. If USE.CTS is clear, the DMB32 will ignore CTS.

Input ML2 (LSTAT<10>) is set if the modem is in test mode.

3.4.11 Sync Modem Control

Sync modem status and control signals are recorded in LPR2<15:0>. (See Section 3.3.35, LPR2).

When a modem status change occurs on the sync channel, the new state is put in LPR2<15:8> and the host is alerted via the sync interrupt. A change of any status bit except SYNC.RI will be recorded in LPR2 as soon as detected, but SYNC.RI must remain steady for at least 30 ms before the DMB32 will recognize a change of state.

When the host writes modem control bits to LPR2<7:0>, it should not write to the byte that contains modem status. Otherwise, changes of modem status may be lost.

The only hardware control between the modem control logic and the receiver and transmitter logic is MODEM.OVERRIDE (see below). Therefore any coordination should be done under program control. MODEM.OVERRIDE is provided to support such coordination.

If MODEM.OVERRIDE is set, transmission and reception can occur regardless of the state of the modem control bits. If MODEM.OVERRIDE is clear, then loss of DCD, DSR, or Receive Clock will cause the DMB32 to abort reception; loss of DSR, CTS, or Transmit Clock will cause transmission to be aborted. In either case, modem status reports and sync interrupts will occur as previously described.

The remaining control and status signals CLOCK.CONTROL, LOOP, and SYNC.TEST.INPUT are used for test purposes.

3.4.12 Selecting Protocols

Synchronous protocols DDCMP, SDLC, HDLC, IBM BISYNC, and GEN BYTE are selected by PROTOCOL<2:0> in LPR2.

CHAPTER 4
TROUBLESHOOTING

4.1 SCOPE

This chapter describes how diagnostics are used to diagnose faults on the DMB32 option. A troubleshooting flowchart is included.

CAUTION

You must wear an anti-static wrist strap connected to an active ground whenever you work on a system with the covers removed, or handle the T1012 module.

4.2 DIAGNOSTICS

There are five types of diagnostic provided for the DMB32.

1. On-board self-test diagnostic
2. Standalone diagnostic: EVDAK (level 3)
3. Online diagnostics: EVDAJ, EVDAL, EVAAA (level 2R)
4. User Environment Test Program (UETP)
5. Data Communications Link Test (DCLT)

As its name suggests, the on-board self-test exists in ROM on the module.

Both level 3 and 2R diagnostics run under the VAX Diagnostic Supervisor (VDS). EVDAK is standalone, while the level 2R diagnostics run online under the VAX/VMS operating system.

UETP is a system exerciser that runs directly under the VAX/VMS operating system.

DCLT is a level 2R diagnostic that tests a data communications link to another processor that is also running DCLT. DCLT can also run with a loopback connector on the modem connection.

4.3 ON-BOARD SELF-TEST DIAGNOSTIC

The self-test diagnostic occupies less than 8 Kwords of the DMB32's firmware ROM. It runs automatically:

- At power-up
- After a hardware or software reset
- When the Node Reset (NRST) bit (VAXBICSR<10>) is set

The self-test performs extensive functional tests of the DMB32 hardware, and then partially initializes the DMB32 before passing control to the functional firmware.

During the tests, the two yellow GO/NOGO LEDs on the T1012 module are turned OFF. They are turned ON again when the option has passed the self-test. If the option does not pass, the LEDs will stay OFF. The GO/NOGO LEDs are driven by the DIAG.FAIL bit (MAINT<11>).

The self-test also reports error and status information to the host through the RX FIFO and some other registers. This information is used by system-based diagnostics such as EVDAK. Diagnostic messages are covered later in this chapter.

Although the self-test is comprehensive, a successful test sequence does not guarantee that all sections of the module are good.

There are three conditions that will cause the self-test sequence to be modified.

- If the SKIP.SELF.TEST bit (MAINT<3>) is set, the self-test will simply initialize the module and pass control immediately to the functional firmware.
- If the FORCE.FAIL bit (MAINT<0>) is set, the self-test will act as if the self-test has failed. It will write an error code into the RX FIFO indicating that it has been forced to fail.
- If the bus signal BI STF L (Self-Test Fast) is asserted a fast self-test will be done. Control will be passed to the functional firmware within 250 ms.

The scope and duration of these variations of the self-test are summarized in Table 4-1.

Table 4-1 Scope and Duration of Self-Test

BI STF L Asserted?	Skip Self-Test?	Force Fail?	Duration	% of Module Tested
x	YES	x	(very short)	–
x	x	YES	(very short)	–
NO	NO	NO	<10 s	50 to 90
YES	NO	NO	<250 ms	15 to 20

x = don't care

4.3.1 Starting the Self-Test

There are several methods by which the host system can start the self-test diagnostic.

- System power-up – Generates a bus reset (via DC LO L) at all nodes (includes resetting BIIC)
- Bus reset – Same as system power-up
- Programmed reset (MAINT<1>) – Does not cause BIIC to self-test, but SKIP.SELF.TEST (MAINT<3>) and FORCE.FAIL (MAINT<0>) bits are examined
- Node Reset (VAXBICSR<10>) – Causes full reset and self-test, including BIIC

Whatever the method of initiation, the diagnostic will always test the state of the bus signal BI STF L (Self-test fast).

If the self-test was caused by power-up, bus reset, or by the NRST bit, the BIIC will be reset and will perform its own self-test. During this operation, the BI.BROKE bit in BICSR<12> is set. It will be cleared after a successful test sequence.

4.3.2 Self-Test Indications and Error Codes

When the self-test begins, it sets the DIAG.FAIL bit (MAINT<11>), and sets the PROGRAMMED.RESET bit (MAINT<1>).

- DIAG.FAIL – Switches OFF the GO/NOGO LEDs and asserts the bus signal BI.BAD.L
- PROGRAMMED RESET – Indicates that the contents of registers are not valid

During the testing of the option, error and status information is transferred to:

- The RX FIFO (RBUF)
- MAINT <0,1,3,8,9,10,11,13,14,15>
- CONFIG <23:0>
- TSMR <31> and <25:0> (GPR0)
- BICSR <12> (the BIIC self-test uses other bits of this register)

The programmed reset is then cleared to show that register contents are valid. If there are no errors, DIAG.FAIL is cleared (GO/NOGO LEDs ON).

A detailed interpretation of the information in the MAINT, CONFIG, TSMR, and BICSR registers is given in Chapter 3, Operation and Programming. The interpretation of the error codes loaded into the RX FIFO is given here.

4.3.2.1 Self-Test Error Codes in the RX FIFO – Error codes that the self-test can load into the RX FIFO are explained in Table 4-2. The table is arranged in numerical order according to the lower word of the error number. This makes the list easier to use for reference but gives a random appearance to the grouping of the tests. The error numbers are actually made up in the following way:

<0>	Always 1
<5:1>	Test number
<7:6>	Error number
<14:8>	Not used, always 0
<15>	Always 1
<20:16>	Channel number or EV code
<30:21>	Not used, always 0
<31>	Always 1

The FIFO may contain up to 21 such codes after completion of the self-test.

Table 4-2 Self-Test Error Codes in the RX FIFO

Error Code (hexadecimal)	Error code meaning
8000 8001	* 68000/RAM Stack R/W Fail
8000 8003	* 68000 microprocessor Fail
8000 8005	* ROM CRC Fail
8000 8007	* Local RAM R/W Fail
8000 8009	* Timer Data Path Fail
8000 800B	* Timer Functionality Fail
8000 800D	* Cable Connector Key Fail
8000 800F	* CASRAM R/W Fail
8000 8011	RX FIFO Alarm Status Fail
8000 8013	TX Action FIFO Alarm Status Fail
8000 8015	Sync TX FIFO Alarm Status Fail
80$$ 8017	Asynchronous Address/Data Path Fail
80$$ 8019	Asynchronous Internal TX/RX Fail
80$$ 801B	Asynchronous External TX/RX Fail
80$$ 801D	Asynchronous Framing Error Fail
80$$ 801F	Asynchronous Parity Error Fail
80$$ 8021	Asynchronous Split Speed Fail
80$$ 8023	Asynchronous Modem Signals Fail
80$$ 8025	Synchronous Data Path Fail
80$$ 8027	Synchronous Internal BOP TX/RX Fail

$$ indicates the channel number; that is: async channels 0 to 7, sync channels A (0), B (1).
%% indicates the EV code obtained from the BIIC via the CGA.
@@ indicates for Digital Internal Use only.
* indicates errors that are FATAL to the self-test.

Table 4-2 Self-Test Error Codes in the RX FIFO (continued)

Error Code (hexadecimal)	Error code meaning
80$$ 8029	Synchronous External V.11 TX/RX Fail
8000 802B	Synchronous Modem Signals Fail
8000 802D	Printer Port Data Output Fail
8000 802F	BIIC/CGA Loopback Timeout or Data Fail
8000 8031	BI/CGA Intra-node Timeout Fail
8000 8033	BI/CGA Intra-node Mask Timeout Fail
8000 8051	RX FIFO Data Fail
8000 8053	TX Action FIFO Data Fail
8000 8055	Sync TX FIFO Data Fail
80$$ 8067	Synchronous Internal COP TX/RX Fail
80$$ 8069	Synchronous External V.10 TX/RX Fail
8000 806D	Printer Port Control Signals Fail
80%% 806F	BIIC/CGA Loopback Master Seq Fail
80%% 8071	BI/CGA Intra-node Master Seq Fail
80%% 8073	BI/CGA Intra-node Mask Master Fail
8000 80A7	Bad Synchronous Interrupt Fail
80$$ 80A9	Synchronous External V.35 TX/RX Fail
80%% 80AF	BIIC/CGA Loopback Slave Seq Fail
80%% 80B1	BI/CGA Intra-node Slave Seq Fail
80%% 80B3	BI/CGA Intra-node Mask Slave Fail
8000 80EF	BIIC Self-Test Fail
8000 80F1	BI/CGA Intra-node Data Error Fail
8000 80F3	BI/CGA Intra-node Mask Data Fail
80@@ 8FFF	Forced Failure

$$ indicates the channel number; that is: async channels 0 to 7, sync channels A (0), B (1).
%% indicates the EV code obtained from the BIIC via the CGA.
@@ indicates for Digital Internal Use only.
* indicates errors that are FATAL to the self-test.

4.3.2.2 Test Summary Register Bit Definitions – With certain self-test detected errors, it is not possible to report via the RX FIFO. As a safeguard against such errors, the DMB32 also makes error reports via the TSMR (also called GPR0).

TSMR<24:00> is used as a bit mask for errors detected by the self-test. If a bit is set, the related error exists. The diagnostic also sets bit <31> to indicate that the data in TSMR is valid. The TSMR register is described in chapter 3, Operation and Programming, Section 3.3.9.

4.4 EVDAK STANDALONE DIAGNOSTIC

EVDAK is a suite of functional verification tests. The tests run standalone under the VAX Diagnostic Supervisor (VDS) V9.0 or later. The supervisor is documented in the *VAX Diagnostic System User's Guide* (EK-VX11D-UG).

EVDAK is intended for:

- Users; to identify failing DMB32s and as a confidence check
- Field Service; to isolate faults and to test installations
- Manufacturing; as a final test
- Final Assembly; as part of a system test

EVDAK has four modes of operation:

1. Internal loopback — In this mode a physical loopback connector is not used. The selected channels are looped back internally. Therefore the line drivers and receivers will not be tested.
2. Manufacturing — This mode is used by DIGITAL to test T1012 modules during manufacture, and is for internal use only.
3. External loopback — In this mode the appropriate loopback connector must be installed on any selected async or sync channel. The connection is made on the H3033 distribution panel. Line drivers and receivers of the looped back channels will be tested.
4. Modem loopback — In this mode only the data lines are tested. Tests can be run with an external loopback connector or a modem on the channel to be tested. If a modem is used it must be looped back manually.

The printer channel can be tested in all available modes (manufacturing mode is not available, and is therefore disregarded in the rest of this chapter). However, a printer is required, and one of the tests requires operator intervention.

EVDAK consists of 33 tests. Any or all of tests 1 to 32 can be selected to run during one pass. Test 33 needs manual intervention and is started with a separate command (see Section 4.4.4). The tests are listed in Table 4-3.

Table 4-3 EVDAK Tests

Test	Function	
1	Device Address test	
2	BIIC Device Address test	
3	Maintenance Modes test	
4	Self-test test	
5	Configuration test	
6	ASYNC PORT TX Enable/Action/TX FIFO test	
7	ASYNC PORT RX FIFO test	
8	ASYNC PORT Interrupt test	
9	ASYNC PORT DMA Start And Abort test	
10	ASYNC PORT Maintenance Mode test	
11	ASYNC PORT Parameters test	
12	ASYNC PORT Split Speed test	(M)
13	ASYNC PORT Error Detection test	(M)
14	ASYNC PORT XON/XOFF test	
15	ASYNC PORT Modem Signals test	
16	ASYNC Port Reset test	
17	SYNC PORT TX Completion FIFO Test	
18	SYNC PORT DMA Start And Abort test	
19	SYNC PORT Maintenance Mode test	
20	SYNC PORT Interrupts test	
21	SYNC PORT Line Protocol test	
22	SYNC PORT Line Parameters test	
23	SYNC PORT Modem Signals test	
24	SYNC PORT Port Reset test	
25	PRINTER PORT DMA Start And Abort test	
26	PRINTER Port Interrupts test	
27	PRINTER PORT ALL Characters test	(V)
28	PRINTER PORT Port Formatting test	(V)
29	PRINTER PORT Prefix And Suffix Characters test	(V)
30	PRINTER PORT Width And Size test	(V)
31	PRINTER PORT Test Patterns	(V)
32	Exercise test	
33	PRINTER ONLINE/OFFLINE test	(I)

(M) Will only run in Manufacturing Mode
(V) Requires visual check of printout
(I) Requires manual intervention and a separate START command

4.4.1 Starting EVDAK

Before EVDAK can be run, the operator must:

- Boot the VDS (EBSAA for VAX-8200, EZSAA for VAX-8800)
- Load EVDAK
- ATTACH and SELECT the DMB32(s) to be tested
- START the diagnostic

These operations are described in the *VAX Diagnostic System User's Guide.*

An example of how to run EVDAK is shown in Example 4-1. In the example, TRACE is set to cause all tests to be reported.

```
DIAGNOSTIC SUPERVISOR.  ZZ-EBSAA- 9.0-325   2-OCT-1985 15:36:22

DS> LOAD EVDAK                ; load the program
DS> SET TRACE                 ; set trace
DS> ATTACH DMB32 HUB TXA 1    ; HUB = linked to VAXBI
                              ; TXA = Device Name as appropriate
                              ;   1 = VAXBI Node ID (hexadecimal)
DS> SELECT TXA                ; select the device for test
DS> START                     ; start the diagnostic
```

Example 4-1 Starting EVDAK

NOTE

If the machine does not have native VAXBI but uses an adapter, this must be attached before the DMB32. For example, on the VAX-8800 the attach sequence might be:

```
DS> ATTACH NBIA HUB NBIA0 0
DS> ATTACH NBIB NBIA NBIB0 0 2
DS  ATTACH DMB32 NBIB0 TXA 4
```

By default, tests 1 to 32 are all run (though tests 12 and 13 will abort without doing anything). Test 33 is run using the \SECTION=MANUAL switch (see Section 4.4.4)

4.4.2 Options

After the start command has been entered, the operator is prompted to select the area to be tested. All possible prompts are illustrated in the example test run shown in Example 4-2.

```
DIAGNOSTIC SUPERVISOR.  ZZ-EBSAA- 9.0-325   2-OCT-1985 15:36:22
DS> LOAD EVDAK
DS> ATTACH
Device type? DMB32
Device Link? HUB
Device Name? TXA
Node_id? 1
DS> SEL TA
DS> START
.. Program: EVDAK, revision 1.0, 33 tests,
   at 15:36:59.45.
Testing: _TA
Test the ASYNC port ? [(YES), YES, NO] NO
Loop back type ? [(INTERNAL), INTERNAL, EXTERNAL, MODEM] MODEM
Lines to be tested ? [(ALL), REVERSE, EVEN, ODD, 0,1,2,...,6,7] 1 2
Line speed ? [(300), 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800,
2000, 2400, 4800, 7200, 9600, 19200, 38400] 1200
Bits per Byte ? [(8), 5, 6, 7, 8] 6
Parity ? [(EVEN), NONE, ODD, EVEN] ODD
Test the SYNC port ? [(YES), YES, NO] YES
Loop back type ? [(INTERNAL), INTERNAL, EXTERNAL, MODEM] MODEM
Test the PRINTER port ? [(YES), YES, NO] YES
Line printer fitted ? [(NO), YES, NO] YES
Printer Page width ? [(132), 0-132(D)] 80
Printer Page length ? [(66), 0-66(D)] 20
Do you want default patterns ? [(YES), YES, NO] NO
Pattern Type ? [(DIAG), DIAG, HORIZ, VERT, CHAR] HORIZ
Enter a test string [ Default is all characters ] THE QUICK BROWN
FOX JUMPED OVER THE LAZY DOG
Repeat pattern ? times [(1), CONT, 1, 2, 3, 4, 5, 6, 7, 8, 9] 6
.. End of run, 0 errors detected, pass count is 1,
   time is 2-OCT-1985 15:39:35.24
DS>
```

Example 4-2 EVDAK Diagnostic Prompts

In the "Lines to be tested" question, up to eight lines can be selected, in any order. The diagnostic will test them in the order that they were entered. "ALL" tests all the lines in ascending order; "REVERSE" tests all the lines in descending order; "EVEN" tests lines 0, 2, 4, and 6; "ODD" tests lines 1, 3, 5, and 7.

When the sync port is tested with an external loopback, the diagnostic will check which of the two 50-way loopback connectors is fitted and will test as appropriate.

Examples of the test patterns for the printer test are shown in Example 4-3. Three different test strings are shown in all four pattern types, the page width is 15 and the page length is 7.

```
    Test String:  default string    THE QUICK BROWN   01

CHAR pattern      AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000

HORIZ pattern     ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101

VERT pattern      AAAAAAAAAAAAAAA   TTTTTTTTTTTTTTT   000000000000000
                  BBBBBBBBBBBBBBB   HHHHHHHHHHHHHHH   111111111111111
                  CCCCCCCCCCCCCCC   EEEEEEEEEEEEEEE   000000000000000
                  DDDDDDDDDDDDDDD                     111111111111111
                  EEEEEEEEEEEEEEE   QQQQQQQQQQQQQQQ   000000000000000
                  FFFFFFFFFFFFFFF   UUUUUUUUUUUUUUU   111111111111111
                  GGGGGGGGGGGGGGG   IIIIIIIIIIIIIII   000000000000000

DIAG pattern      ABCDEFGHIJKLMNO   THE QUICK BROWN   0101010101010101
                  OABCDEFGHIJKLMN    THE QUICK BROW   1010101010101010
                  NOABCDEFGHIJLKM   N THE QUICK BRO   0101010101010101
                  MNOABCDEFGHIJKL   WN THE QUICK BR   1010101010101010
                  LMNOABCDEFGHIJK   OWN THE QUICK B   0101010101010101
                  KLMNOABCDEFGHIJ   ROWN THE QUICK    1010101010101010
                  JKLMNOABCDEFGHI   BROWN THE QUICK   0101010101010101
```

Example 4-3 Printer Test Patterns

4.4.3 EVENT Flags

There are five EVENT flags related to the EVDAK diagnostic. However, they are only used in Manufacturing Mode tests and must not be set in other modes.

4.4.4 Sections

There are two SECTION switches for the START command. These are:

- /SEC=DEFAULT Runs tests 1 to 32
- /SEC=MANUAL Run test 33 only

4.4.5 Error Messages

An error message may indicate that the option is defective or that an illegal parameter has been selected. Error numbers are interpreted in the diagnostic listing EVDAK.LIS. An example of an error message is shown in Example 4-4.

```
******** EVDAK - 1.0 ********
 Pass 1, test 7, subtest 1, error 5, 5-SEP-1985 16:30:04.55
 Hard error while testing TA: ASYNC PORT RX FIFO Failed

Current line number = 05

Expected data was = 0006

Actual data was   = 0007

******** End of Hard error number 5 ********
```

Example 4-4 EVDAK Error Message

4.5 EVDAJ ONLINE DIAGNOSTIC (ASYNC)

This level 2R diagnostic runs online under VMS V4.4 or later. The operator interface is via the VAX Diagnostic Supervisor (VDS) V9.0 or later. EVDAJ provides a confidence check of the async channels of the DMB32. It consists of five tests:

1. Internal data loopback test on selected channels in sequence.
2. Internal DMA data loopback test on selected channels in sequence.
3. Internal DMA data loopback test on selected channels at the same time.
4. External loopback test (using the H3197 loopback) of modem control signals.
5. External data loopback (using the H3197 loopback, or via a modem). If a modem is used, it must be set up manually.

Before running EVDAJ, the user should be aware of the following points:

1. The tests run online, therefore it is essential to define the channels to be tested. The test will not run if a channel which is allocated to another process is selected for testing. Channel allocations can be checked by using the VAX/VMS command SHOW DEVICE/FULL. In Example 4-5, channels 0 and 1 are free and channel 2 is allocated to job control.

```
$ SHOW DEVICE/FULL TXA

Terminal TXA0:, device type VT100, is online, record-oriented
  device, carriage control.

  Error count                    0     Operations completed          118
  Owner process                  ..    Owner UIC                   [1,4]
  Owner process ID       00000000      Dev Prot          S:RWLP,O:MG:,W:
  Reference count                0     Default buffer size            80

Terminal TXA1:, device type VT100, is online, record-oriented
  device, carriage control.

  Error count                    0     Operations completed          113
  Owner process                  ..    Owner UIC                   [1,4]
  Owner process ID       00000000      Dev Prot          S:RWLP,O:MG:,W:
  Reference count                0     Default buffer size            80

Terminal TXA2:, device type VT100, is online, record-oriented
  device, carriage control.

  Error count                    0     Operations completed          113
  Owner process     "Job_control"      Owner UIC                   [1,4]
  Owner process ID       0000008B      Dev Prot          S:RWLP,O:MG:,W:
  Reference count                1     Default buffer size            80
```

Example 4-5 The SHOW DEVICE/FULL Command

2. If test 4 is being run, an H3197 loopback connector must be fitted to the selected channel. By setting EVENT flag 20, the operator will cause the program to halt before each channel is tested. This allows the H3197 to be moved to the next channel to be tested. Another event flag (21) causes the program to halt before each complete DMB32 has been tested.

3. If test 5 is to be run, a modem or an H3197 must be fitted to the selected channel. EVENT flags 20 and 21 also function in this test.

4. If tests 4 or 5 are run without some form of loopback installed, error messages will be generated.

The EVDAJ diagnostic is supported by the online help facility EVDAJ.HLP.

The *VAX Diagnostic System User's Guide* (EK-VX11D-UG) provides instructions on how to load and run programs under the diagnostic supervisor. For details of online tests, refer to the diagnostic listing EVDAJ.LIS, or the help file EVDAJ.HLP.

4.5.1 Starting EVDAJ

Before EVDAJ can be run, VMS must be running and the operator logged into the system maintenance account. The operator must then:

- Allocate the lines to be tested
- Boot the VDS (EBSAA for VAX-8200, EZSAA for VAX-8800)
- Load EVDAJ
- ATTACH and SELECT the DMB32(s) to be tested
- START the diagnostic

These operations are described in the *VAX Diagnostic System User's Guide.*

An example of how to run EVDAJ is shown in Example 4-6. Where appropriate, commands for different host systems are included.

```
$ ALL TXA0                      ; Allocate lines to be tested
$ ALL TXA1                      ;
$ ALL TXA2                      ;
$ RUN EBSAA                     ; For VAX-8200

$ ! RUN EZSAA                   ; For VAX-8800

DIAGNOSTIC SUPERVISOR.  ZZ-EBSAA- 9.0-325   2-OCT-1985 15:36:22
DS> LOAD EVDAJ                  ; Load the DMB32 diagnostic
DS> ATTACH DMB32 HUB TXA 1      ; HUB = linked to VAXBI
                                ; TXA = Device Name as appropriate
                                ;   1 = VAXBI Node ID (hexadecimal)
DS> SELECT TXA                  ; select the device for test
DS> START                       ; start the diagnostic
```

Example 4-6 Starting EVDAJ

NOTE

If the machine does not have native VAXBI but uses an adapter, this must be attached before the DMB32. For example, on the VAX-8800 the attach sequence might be:

```
DS> ATTACH NBIA HUB NBIA0 0
DS> ATTACH NBIB NBIA NBIB0 0 2
DS  ATTACH DMB32 NBIB0 TXA 4
```

4.5.2 Options

After the start command has been entered, the operator is prompted to select the options. There are two questions to be answered, and they are illustrated in the example test run shown in Example 4-7.

```
DIAGNOSTIC SUPERVISOR.  ZZ-EBSAA- 9.0-325   2-OCT-1985 15:36:22
DS> LOAD EVDAJ
DS> ATTACH
Device type? DMB32
Device Link? HUB
Device Name? TXA
Node_id? 1
DS> SEL TXA
DS> START
.. Program: EVDAJ - LEVEL 2R DIAGNOSTIC FOR ASYNC DMB32, revision
1.0, 5 tests, at 15:36:59.45.
Testing: _TXA

Lines to be tested ? [(ALL), REVERSE, EVEN, ODD, 0,1,2,...,6,7] ODD
Line speed ? [(9600), 75, 110, 134, 150, 300, 600, 1200, 1800, 2000,
2400, 4800, 19200] 19200
.. End of run, 0 errors detected, pass count is 1,
   time is  2-OCT-1985 15:37:11.08
DS>
```

Example 4-7 EVDAJ Diagnostic Prompts

In the "Lines to be tested" question, up to eight lines can be selected, in any order. The diagnostic will test them in the order that they were entered. "ALL" tests all the lines in ascending order; "REVERSE" tests all the lines in descending order; "EVEN" tests lines 0, 2, 4, and 6; "ODD" tests lines 1, 3, 5, and 7.

4.5.3 EVENT Flags

Event flags are used to control multi-channel or multi-DMB32 tests. For example:

```
DS> SET EVENT 20
```

Event flag 20 – Functions in tests 4 and 5 only. When flag 20 is set the program halts before each channel is tested. This allows the operator to move the H3197 loopback connector to the channel to be tested next, or to put the modem into local loopback.

Event flag 21 – Functions in all tests. When flag 21 is set, the program halts before each DMB32 is tested.

4.5.4 Sections

There are three SECTION switches for the START command. These are:

- /SEC=DEFAULT Runs tests 1, 2 and 3
- /SEC=EXTERNAL Runs tests 1, 2 and 3, and tests 4 and 5 in external loopback mode
- /SEC=MODEM Runs tests 1, 2 and 3, and test 5 in external loopback mode

4.5.5 Error Messages

If an error is detected during test, the program will output an error message. This may indicate that the option is defective or that an illegal parameter has been selected. Error numbers are interpreted in the diagnostic listing EVDAJ.LIS. An example of an error message is shown in Example 4-8.

```
********  EVDAJ - LEVEL 2R DIAGNOSTIC FOR ASYNC DMB32 - 1.0  *******
 Pass 1, test 4, subtest 0, error 3,  6-SEP-1985 13:23:43.53
 Hard error while testing TXA: TRANSMITTED RTS, EXPECTED CARRIER
AND CTS

 ON LINE: 1
 DEVICE NAME: _TXA1:
 EXPECTED MODEM SIGNALS: 30(X)
 ACTUAL MODEM SIGNALS: 70(X)
 XOR SIGNALS 40(X)                ;RING

********  End of Hard error number 3 ********
```

Example 4-8 EVDAJ Error Message

4.6 EVDAL ONLINE DIAGNOSTIC (SYNC)

This level 2R diagnostic runs online under VMS V4.4 or later. The operator interface is via the VAX Diagnostic Supervisor (VDS) V9.0 or later. EVDAL provides a confidence check of the sync channels of the DMB32 option. It consists of four tests:

1. DDCMP protocol test
2. BISYNC protocol test
3. HDLC protocol test
4. Modem signal test

In tests 1, 2 and 3:

- Data is transferred in groups of 16, 64, 512, and 1000 bytes, using five different data patterns.
- If the 50-way balanced loopback connector H3196 is fitted and the /EXTERNAL section is being run, the test will automatically run twice. Once with V.35 clear and once with V.35 set.

In test 3:

- SDLC is not specifically tested as it is a subset of HDLC.

In test 4:

- The diagnostic detects which loopback connector/adapter cable is connected. The appropriate modem signals are tested.

The EVDAL diagnostic is supported by the online help facility EVDAL.HLP.

The *VAX Diagnostic System User's Guide* (EK-VX11D-UG) provides instructions on how to load and run programs under the diagnostic supervisor. For details of online tests, refer to the diagnostic listing EVDAL.LIS, or the help file EVDAL.HLP.

4.6.1 Starting EVDAL

Before EVDAL can be run, VMS must be running and the operator logged into the system maintenance account. The operator must then:

- Allocate the lines to be tested
- Boot the VDS (EBSAA for VAX-8200, EZSAA for VAX-8800)
- Load EVDAL
- ATTACH and SELECT the DMB32(s) to be tested
- START the diagnostic

These operations are described in the *VAX Diagnostic System User's Guide.*

An example of how to run EVDAL is shown in Example 4-9. Where appropriate, commands for different host systems are included.

```
$ RUN EBSAA                    ; For VAX-8200

$ ! RUN EZSAA                  ; For VAX-8800

DIAGNOSTIC SUPERVISOR.  ZZ-EBSAA- 9.0-325   2-OCT-1985 15:36:22
DS> LOAD EVDAL                 ; Load the DMB32 diagnostic
DS> ATTACH DMB32 HUB SIA 1     ; HUB = linked to VAXBI
                               ; SIA = Device Name as appropriate
                               ;   1 = VAXBI Node ID (hexadecimal)
DS> SELECT SIA                 ; select the device for test
DS> START                      ; start the diagnostic
```

Example 4-9 Starting EVDAL

NOTE

If the machine does not have native VAXBI but uses an adapter, this must be attached before the DMB32. For example, on the VAX-8800 the attach sequence might be:

```
DS> ATTACH NBIA HUB NBIA0 0
DS> ATTACH NBIB NBIA NBIB0 0 2
DS  ATTACH DMB32 NBIB0 TXA 4
```

4.6.2 EVENT Flags

One event flag can be used to control multi-DMB32 tests. For example:

```
DS> SET EVENT 21
```

Event flag 21 – Functions in all tests. When flag 21 is set the program halts before each DMB32 is tested.

4.6.3 Sections

There are three SECTION switches for the START command. These are:

- /SEC=DEFAULT — Runs tests 1,2 and 3 in internal loopback mode. Loopback connectors are NOT REQUIRED in this mode.

- /SEC=EXTERNAL — Runs all tests in external loopback mode. In this mode, a loopback connector IS REQUIRED. Several different loopback connectors can be used with this test:

H3195 50-way loopback	fitted to the
H3196 50-way loopback	distribution panel
H3250 (V.35)	connected via
H3248 (RS-232)	the appropriate
H3198 (RS-422, RS-423)	adapter cable

  To fully test the T1012 module, the diagnostic must be run twice, first with H3195 fitted, and then again with H3196 fitted.

- /SEC=MODEM — Tests 1, 2 and 3 in external loopback mode. This test checks the data lines only. It can be run with any of the loopback arrangements described in /EXTERNAL above, or with a modem installed. If a modem is used, it must be set manually into loopback mode.

4.6.4 Error Messages

If an error is detected during test, the program will output an error message. This may indicate that the option is defective. Error numbers are interpreted in the diagnostic listing EVDAL.LIS. An example of an error message is shown in Example 4-10.

```
  ********* EVDAL - LEVEL 2R DIAGNOSTIC FOR SYNC DMB32 ********
 Pass 1, test 1, subtest 0, error 17, 3-DEC-1985 21:27:12.89
 System fatal error while testing SIA: ERROR OCCURRED DURING WRITE OPERATION

   DEVICE NAME: _SIA0
   IOSB = 0000002C(X)
        = 00010000(X)
   STATUS MESSAGE: %ABORT, abort

******** End of System fatal error number 17 ********
```

Example 4-10 EVDAL Error Message

4.7 EVAAA ONLINE DIAGNOSTIC (Printer)

This level 2R diagnostic runs online under VMS V4.4 or later. The operator interface is via the VAX Diagnostic Supervisor (VDS) V9.0 or later. EVAAA provides a functional test of printers connected to any VAX-11 or VAXBI system (it is not specific to the DMB32).

EVAAA consists of 33 tests for a range of printers. When the diagnostic runs, only those tests relevant to the type of printer connected will be used. A successful test of an attached printer implies that the DMB32 printer port is good.

All EVAAA tests require intervention by the operator or visual inspection of the printout. For details of the tests, refer to the diagnostic listing EVAAA.LIS

4.7.1 Starting EVAAA

Before EVAAA can be run, VMS must be running and the operator logged into the system maintenance account. The operator must then:

- Allocate the lines to be tested
- Boot the VDS (EBSAA for VAX-8200, EZSAA for VAX-8800)
- Load EVAAA
- ATTACH the DMB32(s)
- ATTACH the printer(s)
- SELECT the printer(s) to be tested
- START the diagnostic

These operations are described in the *VAX Diagnostic System User's Guide.*

An example of how to run EVAAA is shown in Example 4-11. Where appropriate, commands for different host systems are included.

```
$ RUN EBSAA                    ; For VAX-8200

$ ! RUN EZSAA                  ; For VAX-8800

DIAGNOSTIC SUPERVISOR.  ZZ-EBSAA- 9.0-325   2-OCT-1985 15:36:22
DS> LOAD EVAAA                 ; Load the DMB32 diagnostic
DS> ATTACH DMB32 HUB LIA 1     ; HUB = linked to VAXBI
                               ; LIA = Device Name as appropriate
                               ;   1 = VAXBI Node ID (hexadecimal)
DS> ATTACH LN01 LIA LIA0       ; LN01 = printer to be tested
                               ; LIA = linked to LIA
                               ; LIA1 = device name
DS> SELECT LIA0                ; select the device for test
DS> START                      ; start the diagnostic
```

Example 4-11 Starting EVAAA

NOTE

If the machine does not have native VAXBI but uses an adapter, this must be attached before the DMB32. For example, on the VAX-8800 the attach sequence might be:

```
DS> ATTACH NBIA HUB NBIA0 0
DS> ATTACH NBIB NBIA NBIB0 0 2
DS  ATTACH DMB32 NBIB0 TXA 4
```

4.7.2 Error Messages

If an error is detected during test, the program will output an error message. This may indicate that the option is defective. Error numbers are interpreted in the diagnostic listing EVAAA.LIS. An example of an error message is shown in Example 4-12.

```
********  EVAAA LINE PRINTER DIAGNOSTIC - 5.6  ********
 Pass 1, test 1, subtest 0, error 4, 13-JUN-1985 08:42:26:64
 System fatal error while testing LCA0: SYSTEM ERROR ON WRITE

********  End of system fatal error number 4 ********

 ..Aborted program at pass 1, test 1, subtest 0, PC 000003EF.
```

Example 4-12 EVAAA Error Message

4.8 USER ENVIRONMENT TEST PROGRAM (UETP)

After the DMB32 has successfully passed the relevant diagnostic tests, the UETP system exerciser should be run to check for interaction problems between DMB32 and other options.

4.9 DATA COMMUNICATIONS LINK TEST (DCLT)

If the on-board self-tests detects no faults, DCLT can be used to test the external line.

DCLT is a diagnostic program that tests communications links. It is used to isolate errors to the local communications device, the modem, the physical line, or the remote communications device. Each DCLT diagnostic is written for the specific option on which it runs, but a DCLT diagnostic can communicate with any other DCLT diagnostic via a communications link. For this test, the system at each end of the link must have DCLT loaded and running.

4.10 FIELD REPLACEABLE UNITS (FRUs)

CAUTION

You must wear an anti-static wrist strap connected to an active ground whenever you work on a system with the covers removed, or handle the T1012 module.

The T1012 module is supplied in protective anti-static packaging. Do not remove the module from its packaging until you are about to install it.

There is no preventive maintenance for the DMB32. Corrective maintenance is based on identifying and replacing a defective FRU. The FRUs for DMB32 are:

- T1012 module
- 17-00740-xx ribbon cable
- H3033 distribution panel
- 12-22246-01 transition header assembly

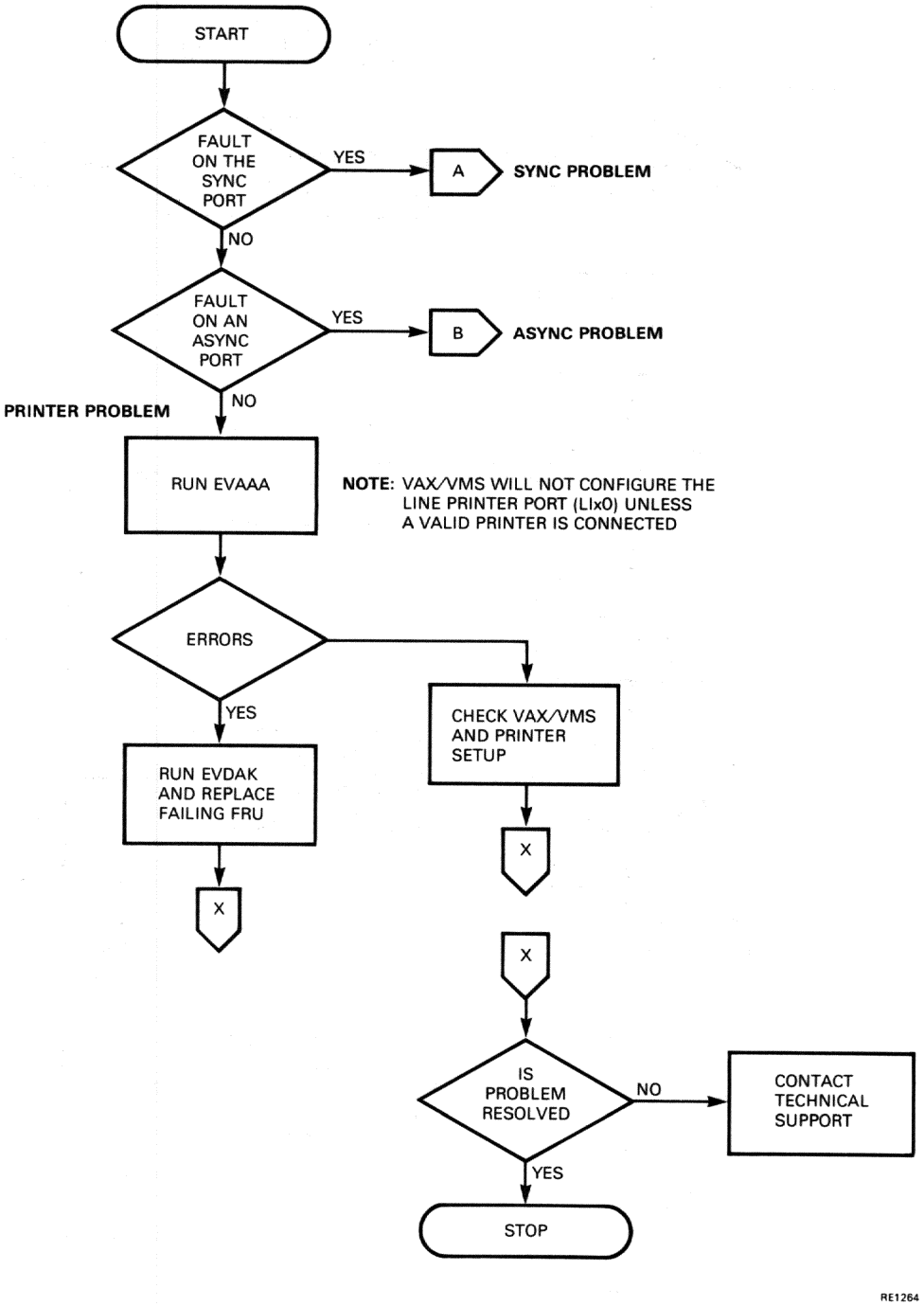
There are six 17-00740-xx ribbon cables on each DMB32 option, but the diagnostics cannot isolate a fault to a single cable. If a fault is isolated to the ribbon cables, Table 4-4 identifies which cable carries which set of signals, and can be used to determine which cable is likely to be faulty. Refer to Chapter 1, Table 1-1 to relate the CCITT circuit numbers given in Table 4-4 to their EIA equivalents.

Table 4-4 Ribbon Cable Signal Distribution

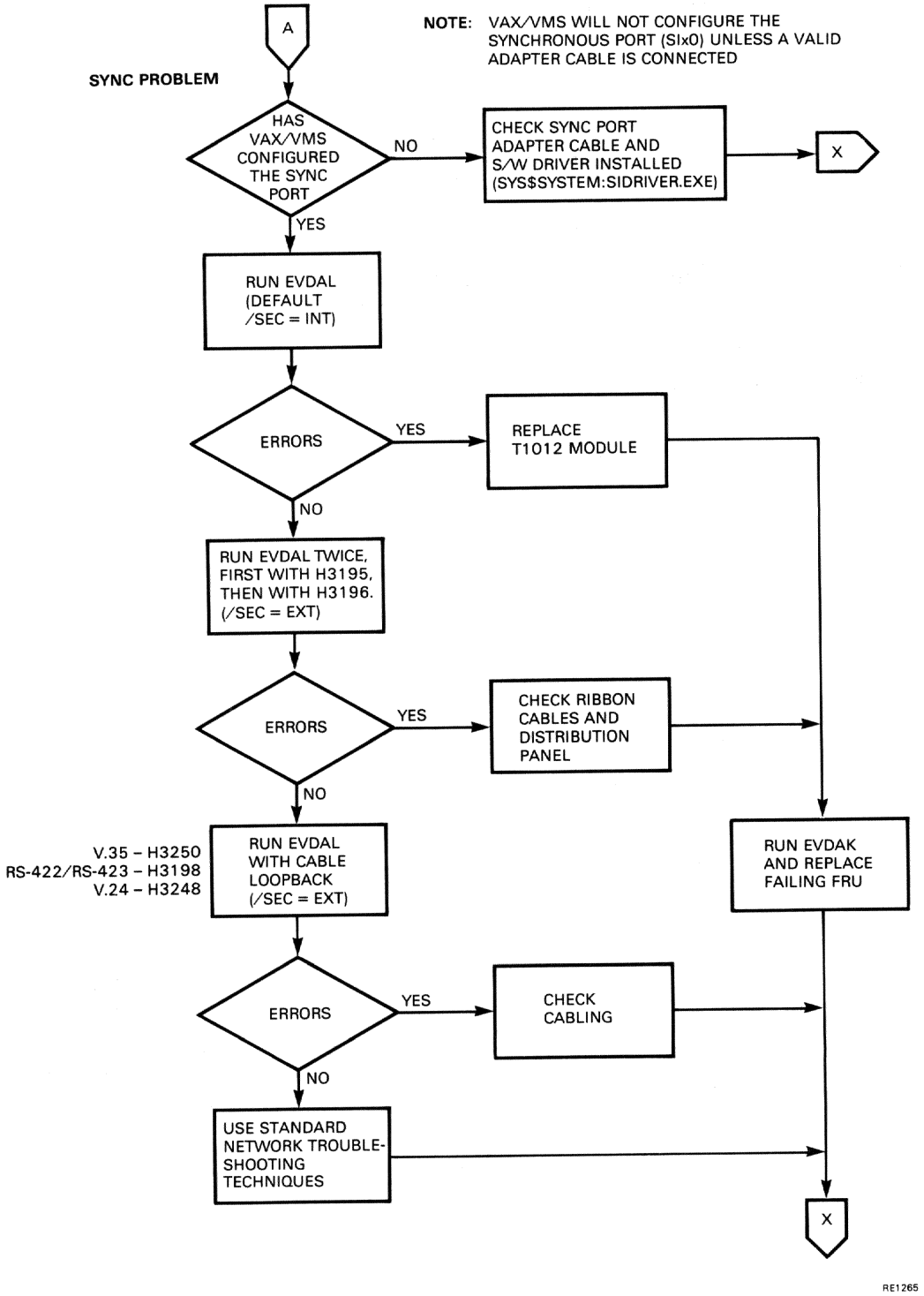
Cable VAXBI Zone	H3033 Socket	Signals Carried
C-1	J10	Async: channels 0 and 2; Printer: DAT<2:1>, ONLINE
C-2	J13	Async: channels 1 and 3; Cable Code <3:0>
D-1	J11	Async: channels 4 and 5; Printer: DAT<5>, DAT<3>
D-2	J14	Async: channels 5 and 7
E-1	J12	Sync: GND, circuit 107, 111, 114; V.10 circuit 104; V.11 circuit 108; V.35 circuit 103, 113, 114, 115; Printer: DAT<4>, DAT<0>, DEMAND, STROBE
E-2	J15	Sync: circuit 104, 106, 109, 115, 125, 142; V.10 circuit 103, 113; V.11 circuit 103, 105, 113; V.35 circuit 104; Printer: DAT<7:6>, CONN, DAVFU

Depending on local maintenance practice, adapter cables, extension cables and modems may also be considered as FRUs.

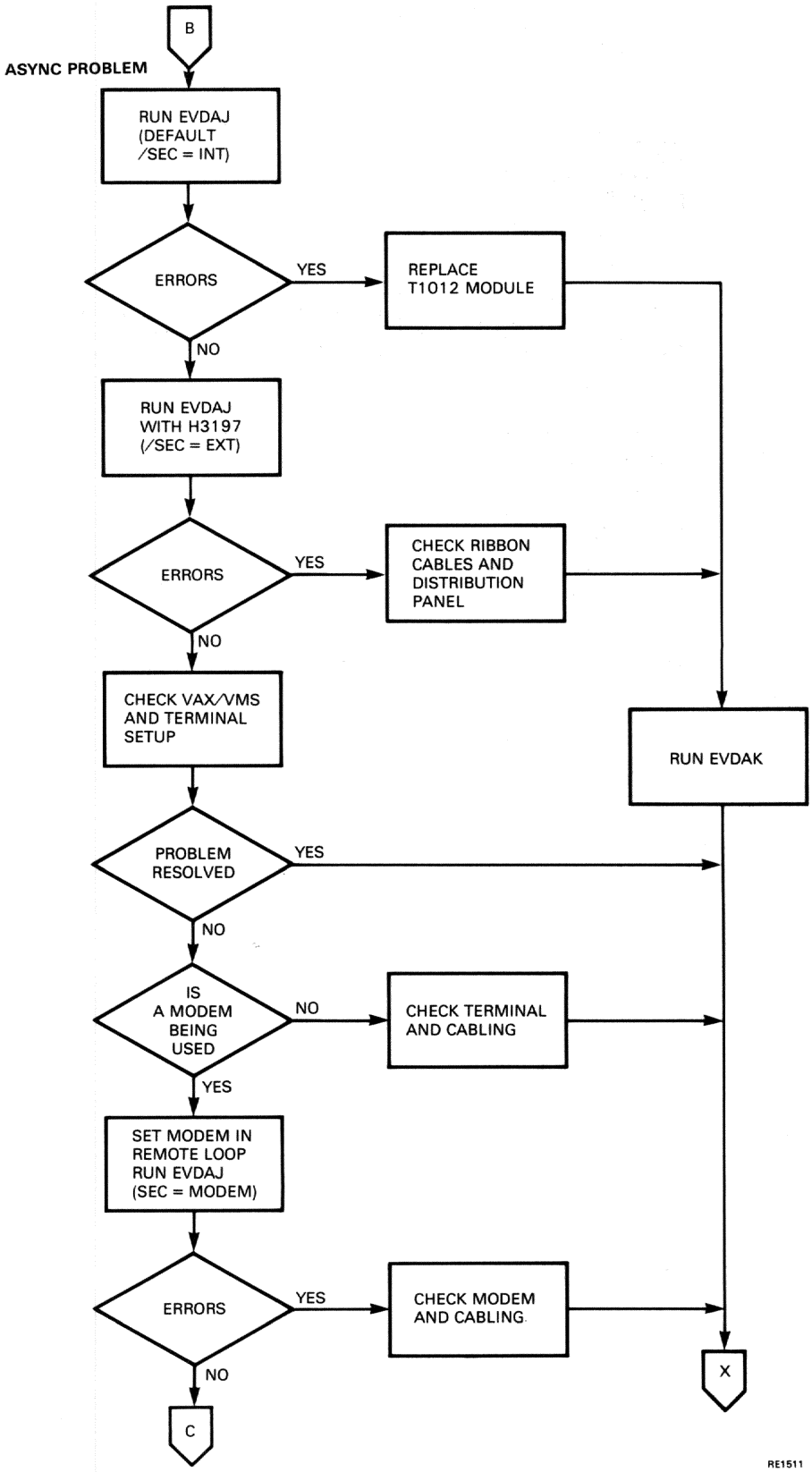
4.11 TROUBLESHOOTING FLOWCHART



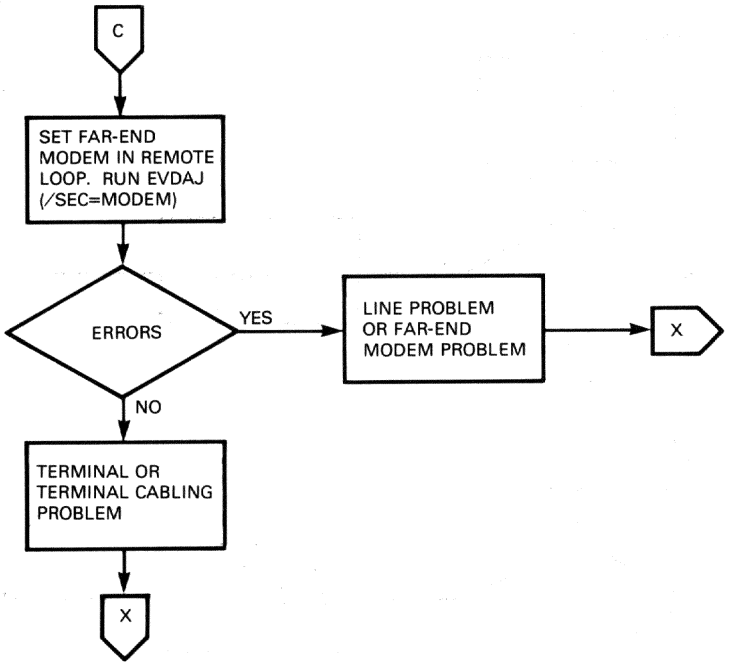
TROUBLESHOOTING FLOWCHART (continued)



TROUBLESHOOTING FLOWCHART (continued)



TROUBLESHOOTING FLOWCHART (continued)



APPENDIX A
GLOSSARY

This glossary defines terms used to describe the VAXBI bus, the BIIC, and the DMB32 option.

Adapter
A node that interfaces other buses, communication lines, or peripheral devices to the VAXBI bus.

Asserted
To be in the "true" state.

Asynchronous
A method of serial data transmission in which data is preceded by a start bit and followed by a stop bit. The receiver provides the intermediate timing to identify the data bits.

Auto-answer
Facility of a modem or terminal to automatically answer a call.

Auto-flow
Automatic flow control. Method by which the DMB32 controls the flow of data by means of special characters within the data stream.

Backward Channel
Channel which transmits in the opposite direction to the usual data flow. Normally used for supervisory or control signals.

Base Address
The address of the CSR.

BCI
VAXBI chip interface; a synchronous interface bus that provides for all communication between the BIIC and the user interface.

BIIC
Backplane interconnect interface chip; a chip that serves as a general purpose interface to the VAXBI bus.

BIIC CSR Space
The first 256 bytes of the 8 Kbyte nodespace, which is allocated to the BIIC's internal registers. See also **Nodespace.**

Bus Adapter
A node that interfaces the VAXBI bus to another bus.

CCITT
Comite Consultatif International de Telephonie et de Telegraphie. An international standards committee for telephone, telegraph and data communications networks.

Dataset
See **Modem**

Deasserted
To be in the "false" state.

Decoded ID
The node ID expressed as a single bit in a 16-bit field.

Device Type
A 16-bit code that identifies the node type. This code is contained in the BIIC's Device Register.

DIL
Dual-In-Line. The term describes ICs and components with two parallel rows of pins.

DMA
Direct Memory Access. Method which allows a bus master to transfer data to/from system memory without using the the host CPU.

DMA adapter
An adapter that directly performs block transfers of data to and from memory.

DUART
Dual Universal Asynchronous Receiver Transmitter. IC used for transmission and reception of serial asynchronous data on two channels.

Duplex
Method of transmitting and receiving on the same channel at the same time.

EIA
Electrical Industries of America. American organization with the same function as CCITT.

EMC
Electro-Magnetic Compatibility. The term denotes compliance with field-strength, susceptibility and static discharge standards.

Encoded ID
The node ID expressed as a 4-bit binary number. The encoded ID is used for the master's ID transmitted during an imbedded ARB cycle.

Even Parity
The parity line is asserted if the number of asserted lines in the data field is an odd number.

FCC
Federal Communications Commission. American organization which regulates and licenses communications equipment.

FIFO
First In First Out. The term describes a register or memory from which the oldest data is removed first.

Floating Address
CSR address assigned to an option which does not have a fixed address allocated. The address is dependent upon other floating address devices connected to the bus.

Floating vector
Interrupt vector assigned to an option which does not have a fixed vector allocated. The vector is dependent upon other floating vector devices connected to the bus.

FRU
Field Replaceable Unit

GO/NOGO
Test or indicator which defines an 'error' or 'no error' condition only.

H
Designates a high-voltage logic level (that is, the logic level closest to Vcc). Contrast with **L**.

IC
Integrated Circuit

Interrupt Vector
In VAX/VMS systems, an unsigned binary number used as an offset into the system control block. The system control block entry pointed to by the VAXBI interrupt vector contains the starting address of an interrupt-handling routine. (The system control block is defined in the *VAX-11 Architecture Reference Manual*.)

I/O
Input/Output

L
Designates a low-voltage logic level (that is, the logic level closest to ground). Contrast with *H*.

LSB
Least Significant Bit

LSI-11 bus
Another name for the Q-bus

Master
The node that gains control of the VAXBI bus and initiates a VAXBI or loopback transaction. See also **Pending Master**.

Master Port
Those BCI signals used to generate VAXBI or loopback transactions.

Master Port Transaction
Any transaction initiated as a result of a master port request.

Microcomputer
An IC which contains a microprocessor and peripheral circuitry such as memory, I/O ports, timers, and **UARTs**.

Modem
The word is a contraction of MOdulator DEModulator. A modem interfaces a terminal to a transmission line. A modem is sometimes called a dataset.

Module
A single VAXBI card that attaches to a single VAXBI connector.

MSB
Most Significant Bit

Multiplexer
A circuit which connects a number of lines to one line.

Node
A VAXBI interface that occupies one of sixteen logical locations on a VAXBI bus. A VAXBI node consists of one or more VAXBI modules.

Node ID
A number that identifies a VAXBI node. The source of the node ID is an ID plug attached to the backplane.

Node Reset
A sequence that causes an individual node to be initialized; initiated by the setting of the Node Reset bit in the VAXBI Control and Status Register.

Nodespace
An 8 Kbyte block of I/O addresses that is allocated to each node. Each node has a unique nodespace based on its node ID.

Null Modem
A cable which allows two terminals which use modem control signals to be connected together directly. Only possible over short distances.

Odd Parity
The parity line is asserted if the number of asserted lines in the data field is an even number.

PCB
Printed Circuit Board

Power-down/Power-up Sequence
The sequencing of the BI AC LO L and BI DC LO L lines upon the loss and restoration of power to a VAXBI system. See also **System Reset.**

Protocol
Set of rules which define the control and flow of data in a communications system.

PSTN
Public Switched Telephone Network

Q-bus
Global term for a specific DIGITAL bus on which the address and data are multiplexed.

Q22, Q18 and Q16
Terms used to define 22-, 18- and 16-bit address versions of Q-bus.

RAM
Random Access Memory

RESERVED code
A code reserved for use by DIGITAL.

RESERVED field
A field reserved for use by DIGITAL. The node driving the bus must ensure that all VAXBI lines in the RESERVED field are deasserted. Nodes receiving VAXBI data must ignore RESERVED field information. This requirement provides for adding functionality to future VAXBI node designs without affecting compatibility with present designs. Example: The BI D<31:0> L and BI I<3:0> L lines during the third cycle of an INTR transaction are RESERVED fields.

Reset Module
In a VAXBI system, the logic that monitors the BI RESET L line and any battery backup voltages and that initiates the system reset sequence.

Resetting Node
The node that asserts the BI RESET L line.

RFI
Radio Frequency Interference

ROM
Read Only Memory

Slave
A node that responds to a transaction initiated by a node that has gained control of the VAXBI bus (the master).

Slave Port
Those BCI signals used to respond to VAXBI and loopback transactions.

Split-speed
Facility of a data communications channel which can transmit and receive at different data rates at the same time.

System Reset
An emulation of the power-down/power-up sequence that causes all nodes to initialize themselves; initiated by the assertion of the BI RESET L line.

Transaction
The execution of a VAXBI command. The term "transaction" includes both VAXBI and loopback transactions.

UART
Universal Asynchronous Receiver Transmitter. **IC** used for transmission and reception of serial asynchronous data on a channel.

UNDEFINED Field
A field that must be ignored by the receiving node(s). There are no restrictions on the data pattern for the node driving the VAXBI bus. Example: The BI D<31:0> L and BI I<3:0> L lines during read STALL data cycles and vector STALL data cycles are UNDEFINED fields.

User Interface
All node logic exclusive of the BIIC.

User Interface CSR Space
That portion of each nodespace allocated for user interface registers. The user interface CSR space is the 8 Kbyte nodespace minus the lowest 256 bytes, which comprise the BIIC CSR space.

VAX Interrupt Priority Level (IPL)
In VAX/VMS systems, a number between 0 and 31 that indicates the priority level of an interrupt with 31 being the highest priority. When a processor is executing at a particular level, it accepts only interrupts at a higher level, and on acceptance starts executing at that higher level.

VAXBI Primary Interface
The portion of a node that provides the electrical connection to the VAXBI signal lines and implements the VAXBI protocol; for example, the BIIC.

VAXBI Request
A request for a VAXBI transaction from the **Master Port** interface that is asserted on the BCI RQ<1:0> L lines.

VAXBI System
All VAXBI cages, VAXBI modules, reset modules, and power supplies thhat are required to operate a VAXBI bus. A VAXBI system can be a subsystem of a larger computer system.

VAXBI Transaction
A transaction in which information is transmitted on the VAXBI signal lines.

Window Space
A 256 Kbyte block of I/O addresses allocated to each node based on node **ID** and used by bus adapters to map VAXBI transactions to other buses.

XOFF
Control code (23 octal) used to disable a transmitter. Special hardware or software is needed for this function.

XON
Control code (21 octal) used to enable a transmitter which has been disabled by an **XOFF** code.

Digital Equipment Corporation • Bedford, MA 01730