
CXA16/CXB16

EK-CAB16-TM-002

CXA16/CXB16

Technical Manual

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, December 1986
2nd Edition, September 1989

Printed in U.S.A.

Book production was done by Educational Services Development and Publishing in Merrimack, NH.

CONTENTS

PREFACE

CHAPTER 1 INTRODUCTION

1.1 SCOPE 1-1
1.2 OVERVIEW 1-1
1.2.1 General Description 1-1
1.2.1.1 Self-Test Facility 1-3
1.2.1.2 Diagnostic Programs 1-3
1.2.1.3 Preventing Data Loss 1-3
1.2.2 Physical Description 1-3
1.2.2.1 On-Board Switchpacks 1-5
1.2.3 Configurations 1-5
1.2.4 Connections 1-7
1.3 SPECIFICATION 1-8
1.3.1 Environmental Conditions 1-8
1.3.2 Electrical Requirements 1-8
1.3.2.1 Q-bus Loads 1-8
1.3.3 Performance 1-9
1.3.3.1 Data Rates 1-9
1.3.3.2 Throughput 1-9
1.4 SERIAL INTERFACES 1-9
1.4.1 Interface Standards 1-9
1.4.2 Speed and Distance Considerations 1-9
1.4.3 Line Receivers 1-12
1.4.4 Line Transmitters 1-12
1.5 FUNCTIONAL DESCRIPTION 1-13
1.5.1 General 1-13
1.5.2 CXA16/CXB16 Main Functions 1-15
1.5.2.1 Transmission 1-15
1.5.2.2 Reception 1-15
1.5.3 Control Chip 1-15
1.5.4 OCTART Chip 1-16

CHAPTER 2 INSTALLATION

2.1 SCOPE 2-1
2.2 UNPACKING AND INSPECTION 2-1
2.3 PREPARING THE CXA16/CXB16 MODULE 2-2
2.3.1 Address and Vector Assignment 2-2
2.3.2 DHV11 or DHU11 Programming Mode Selection 2-3
2.3.3 Setting the Address Switches 2-4
2.3.4 Setting the Vector Switches 2-5

2.4 BUS CONTINUITY 2-6
2.4.1 Bus Grant Continuity Jumpers 2-6
2.5 PRIORITY SELECTION 2-7
2.5.1 Recommendations 2-7
2.5.2 DMA Request Priority 2-7
2.6 CXA16/CXB16 INSTALLATION 2-8
2.7 INSTALLATION TESTING 2-11
2.7.1 Installation Tests on MicroPDP-11 Systems 2-11
2.7.2 Testing in MicroVAX II Systems 2-12
2.8 LOOPBACK TEST CONNECTORS 2-12
2.8.1 H3101 Loopback Connector 2-12
2.8.2 H3103 Loopback Connector 2-14

CHAPTER 3 PROGRAMMING

3.1 SCOPE 3-1
3.2 REGISTERS 3-1
3.2.1 Register Access 3-1
3.2.2 Register Bit Definitions 3-4
3.2.2.1 Control and Status Register (CSR) 3-4
3.2.2.2 Receive Buffer (RBUF) 3-9
3.2.2.3 Transmit-Character Register (TXCHAR): DHV11 mode only 3-12
3.2.2.4 Receive Timer Register (RXTIMER): DHU11 mode only .. 3-13
3.2.2.5 Line-Parameter Register (LPR) 3-14
3.2.2.6 FIFO Data Register (FIFODATA): DHU11 mode only 3-17
3.2.2.7 FIFO Size Register (FIFOSIZE): DHU11 mode only..... 3-19
3.2.2.8 Line-Status Register 3-20
3.2.2.9 Line-Control Register 3-21
3.2.2.10 Transmit Buffer Address Register Number 1 (TBUFFAD1) 3-25
3.2.2.11 Transmit Buffer Address Register Number 2 (TBUFFAD2) 3-26
3.2.2.12 Transmit DMA Buffer Counter (TBUFFCT) 3-28
3.3 PROGRAMMING FEATURES 3-29
3.3.1 Initialization 3-29
3.3.2 Configuration 3-30
3.3.3 Transmitting 3-30
3.3.3.1 DMA Transfers 3-31
3.3.3.2 Programmed I/O (DHV11 Mode) 3-32
3.3.3.3 Programmed I/O (DHU11 Mode) 3-32
3.3.4 Receiving 3-33
3.3.5 Interrupt Control 3-33
3.3.6 Auto XON and XOFF 3-34
3.3.7 Error Indication 3-36
3.3.8 Maintenance Programming 3-37
3.3.9 Diagnostic Codes 3-37
3.3.9.1 Self-Test Diagnostic Codes 3-37
3.3.9.2 Interpretation of Self-Test Codes 3-37
3.3.9.3 Skipping Self-Test 3-38
3.3.9.4 Background Monitor Program (BMP) 3-39

3.4 PROGRAMMING EXAMPLES 3-39
3.4.1 Resetting the CXA16/CXB16 3-40
3.4.2 Configuration 3-41
3.4.3 Transmitting 3-42
3.4.3.1 Programmed Transfer (DHU11 Mode) 3-42
3.4.3.2 Single-Character Programmed Transfer (DHV11 Mode) ... 3-43
3.4.3.3 DMA Transfer 3-44
3.4.3.4 Aborting a Transmission 3-46
3.4.4 Receiving 3-47
3.4.5 Auto XON and XOFF 3-49
3.4.6 Checking Diagnostic Codes 3-51

CHAPTER 4 TROUBLESHOOTING

4.1 SCOPE 4-1
4.2 TROUBLESHOOTING 4-1
4.2.1 Preventive Maintenance 4-1
4.2.2 Troubleshooting Procedures 4-1
4.3 INTERNAL DIAGNOSTICS 4-4
4.3.1 Self-Test 4-4
4.3.2 Background Monitor Program (BMP) 4-4
4.4 MicroPDP-11 DIAGNOSTICS 4-5
4.4.1 User-Mode Diagnostics 4-5
4.4.1.1 Running User-Mode Tests 4-5
4.5 MicroVAX II DIAGNOSTICS 4-5
4.5.1 User-Mode Tests 4-5
4.6 MicroPDP-11 SERVICE-MODE DIAGNOSTICS 4-5
4.6.1 MicroPDP-11 Functional Diagnostic 4-5
4.6.1.1 Test Summaries 4-6
4.6.1.2 Loopback Connectors 4-10
4.6.2 DECX11 Object Module 4-10
4.7 MicroVAX II SERVICE-MODE DIAGNOSTICS 4-11
4.7.1 Configuration Tests 4-11
4.7.2 Field Service Functional Tests 4-11
4.7.3 Field Service Exerciser Tests 4-12
4.7.4 Utilities 4-12
4.8 FIELD-REPLACEABLE UNITS (FRUs) 4-13

CHAPTER 5 TECHNICAL DESCRIPTION

5.1 SCOPE 5-1
5.2 OVERVIEW 5-1
5.3 CXA16/CXB16 FUNCTIONAL BLOCKS 5-3
5.3.1 RAM 5-3
5.3.2 OCTART 5-5
5.3.2.1 Control-Chip Interface 5-6
5.3.2.2 Line-Parameter Register 5-6
5.3.2.3 Loop-Around Control 5-6
5.3.2.4 Baud-Rate Generator 5-6

5.3.3 Control Chip 5-7
5.3.3.1 Timer and RAM Test Sequencer 5-7
5.3.3.2 Q-bus Interface (QIF) 5-7
5.3.3.3 DMA Slow Sequencer (DMS) 5-8
5.3.3.4 DMA Fast Sequencer (DMF) 5-8
5.3.3.5 Data I/O Sequencer (DIO) 5-8
5.3.3.6 UART Fast Sequencer (UAF) 5-8
5.3.3.7 UART Slow Sequencer (UAS) 5-9
5.3.3.8 RAM Arbitrator (RARB) 5-10
5.3.4 Switchpack and Shift Registers 5-10
5.3.5 Q-bus Drivers and Receivers 5-13
5.3.6 Line Interfaces 5-13
5.3.7 Power Converter (CXA16 Only) 5-13
5.4 DATA FLOW 5-15
5.4.1 Data Flow for Character Reception 5-15
5.4.2 Data Flow for Character Transmission 5-15
5.4.2.1 DMA Operation 5-15
5.4.2.2 Programmed I/O Operation 5-16

APPENDIX A CXA16/CXB16 Q-BUS CONNECTIONS

APPENDIX B FLOATING ADDRESSES

B.1 FLOATING DEVICE ADDRESSES B-1
B.2 FLOATING VECTORS B-4

APPENDIX C AUTOMATIC FLOW CONTROL

C.1 OVERVIEW C-1
C.2 CONTROL OF TRANSMITTED DATA C-1
C.3 CONTROL OF RECEIVED DATA C-3
C.3.1 Flow Control by the Level of the Receive FIFO C-3
C.3.2 Flow Control by Program Initiation C-5
C.3.3 Mixing the two Types of Received-Data Flow Control C-6

APPENDIX D GLOSSARY OF TERMS

D.1 SCOPE D-1
D.2 GLOSSARY D-1

APPENDIX E CONTROL CHIP AND OCTART

E.1 SCOPE E-1
E.2 CONTROL CHIP E-2
E.3 OCTART E-4

INDEX

FIGURES

Figure No.	Title	Page
1-1	CXA16/CXB16 Module Layout	1-4
1-2	Example of CXA16 Configuration	1-6
1-3	CXA16/CXB16 Connections	1-7
1-4	CXA16/CXB16 Functional Block Diagram	1-14
2-1	Location of the Switchpacks	2-3
2-2	Setting the Device Address	2-4
2-3	Setting the Vector Address	2-5
2-4	Bus Grant Continuity Flow	2-6
2-5	CXA16/CXB16 Installation	2-10
2-6	H3101 Loopback Test Connector	2-13
2-7	Line Loopback Test Connector	2-14
4-1	Cable Loopbacks	4-3
5-1	CXA16/CXB16 Block Diagram	5-3
5-2	RAM Assignment	5-4
5-3	OCTART	5-5
5-4	Switchpack and Shift Register Logic	5-11
5-5	Power Converter	5-14
5-6	Receive Character Data Flow	5-17
5-7	Transmit Character Data Flow	5-18
C-1	Transmitted Data Flow Control	C-2
C-2	Receive FIFO-Level Flow Control	C-4
C-3	Program-Initiated Flow Control	C-5

TABLES

Table No.	Title	Page
1-1	Maximum Distance Guidelines for CXA16/CXB16	1-10
1-2	Serial Line Connections for the 36-pin Connector	1-13
3-1	CXA16/CXB16 Registers in DHV11 Mode	3-2
3-2	CXA16/CXB16 Registers when in DHU11 Mode	3-3
3-3	Data Rates	3-17
3-4	CXA16/CXB16 Self-Test Error Codes	3-37
A-1	CXA16/CXB16 Q-bus Connections	A-1
B-1	Floating Device Address Assignments	B-1
B-2	Floating Vector Address Assignments	B-4
E-1	Interface Signals	E-2
E-2	OCTART Signals	E-5

PAGE VIII INTENTIONALLY LEFT BLANK

PREFACE

The CXA16/CXB16 Technical Manual provides reference information on physical layout, system configuration, installation and testing, programming characteristics, and maintenance; it also includes a technical description of the CXA16/CXB16. There is a glossary of technical terms generally used in DIGITAL technical manuals, and a name index for easy topic reference. The Technical Manual is divided into five chapters as follows:

CHAPTER 1 INTRODUCTION. This chapter gives a physical description of the CXA16/CXB16, explains how it can be configured, and it explains how it interfaces with the system bus and serial data lines.

CHAPTER 2 INSTALLATION. This chapter describes how to install a CXA16/CXB16 option, with detailed information on device and vector address selection, backplane positioning, cables and connectors, and testing after installation.

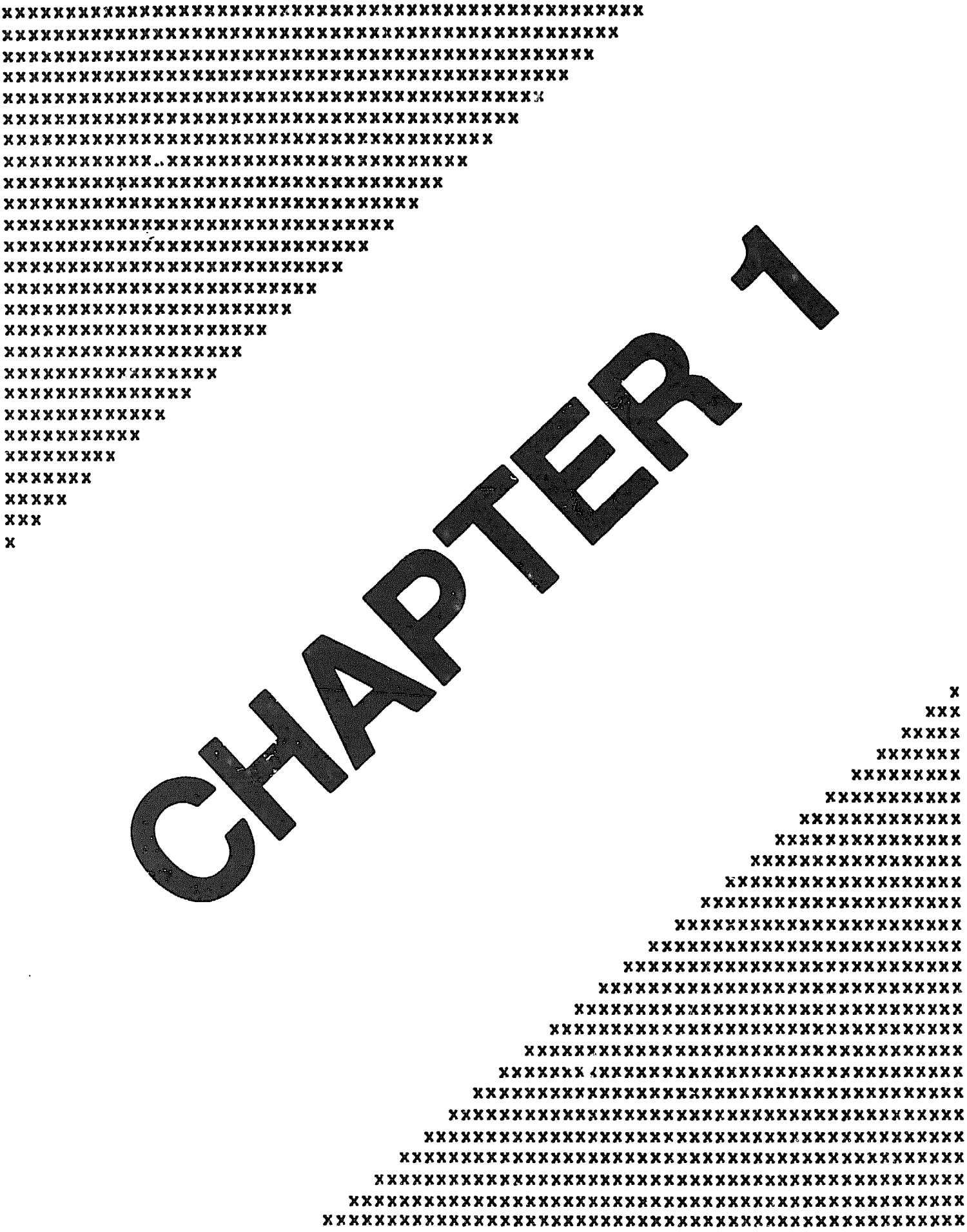
CHAPTER 3 PROGRAMMING. This chapter describes CXA16/CXB16 control registers. Some programming examples are also included.

CHAPTER 4 TROUBLESHOOTING. This chapter explains maintenance strategy and how to use diagnostic programs to locate a faulty module. A simple troubleshooting flowchart is included.

CHAPTER 5 TECHNICAL DESCRIPTION. This chapter gives a technical description of the CXA16/CXB16 asynchronous multiplexer. It is assumed that you have some knowledge of Q-bus operations.

APPENDICES. These expand on topics discussed in this manual:

APPENDIX A - CXA16/CXB16 BUS CONNECTIONS
APPENDIX B - FLOATING ADDRESSES
APPENDIX C - AUTOMATIC FLOW CONTROL
APPENDIX D - GLOSSARY OF TERMS
APPENDIX E - CONTROL CHIP AND OCTART


CHAPTER 1

CHAPTER 1
INTRODUCTION

1.1 SCOPE

This chapter gives an overview of the CXA16/CXB16 asynchronous multiplexer, describes the facilities it offers, and defines its physical parameters and electrical requirements.

1.2 OVERVIEW

1.2.1 General Description

The CXA16/CXB16 is a serial-line interface which provides 16 full-duplex serial data channels for use in BA213/BA214-based Q-bus systems.

The CXA16/CXB16 can be used in many applications, including data concentration, real-time processing, and interactive terminal handling. It has two programming modes: DHV11 and DHU11. The register sets in these two modes are compatible with those of the DHV11 and DHU11 respectively. The preferred mode of operation is the DHU11 mode. The main features of the CXA16/CXB16 are as follows.

- Sixteen full-duplex asynchronous data-only channels
- The CXA16 supports DEC423 connections, and the CXB16 supports DEC422 connections

NOTE

DEC422 is a term used in this manual to indicate a data-leads-only implementation of the RS-422-A standard. DEC423 is a term used in this manual to indicate a data-leads-only implementation of the RS-423-A standard.

- For each line: DMA transfers, and program transfers to a 1-character transmit buffer in DHV11 mode, or program transfers to a 64-character transmit FIFO in DHU11 mode
- A 256-entry FIFO buffer for received characters, and diagnostic information

- For CXA16, the electrical characteristics and signaling standards are compatible with RS-423-A. As the physical interface does not use a 37-way subminiature D-type connector, it does not comply with this section of the RS-449 standards

- For CXB16, the electrical characteristics and signaling standards are compatible with RS-422-A. As the physical interface does not use a 37-way subminiature D-type connector, it does not comply with this section of the RS-449 standards

- The transmit and receive baud rates for each line can be individually programmed

- A total module throughput of 50,000 characters per second, using 8-bit characters, with all channels operating at 38.4 kbaud for both character reception and transmission

- Automatic flow control of transmitted data, with XON/XOFF character reporting through the receive FIFO, if enabled

- Self-test and background monitor testing

- Switch functions for selecting:

  - Addresses and vectors

  - DHV11 or DHU11 programming mode

- All the other functions are selected by program

- The module handle forms part of the system bulkhead. Signal distribution is through two 36-way connectors

- Transient surge suppressors on all serial lines for static discharge and surge protection.

The CXA16/CXB16 is a data device and is not intended for connection to modems or other Wide Area Network equipment. Note that, although the two program modes are referred to as DHV11 and DHU11, the modem control signals supported by these devices are NOT supported by the CXA16/CXB16.

1.2.1.1 Self-Test Facility -- The CXA16/CXB16 incorporates self-test sequencers which operate independently of the host. They test the device on power-up or initialization, and report diagnostic information to the host.

1.2.1.2 Diagnostic Programs -- A full range of diagnostic programs is available. These run under the PDP-11 diagnostic supervisor or MicroVAX II maintenance system. Diagnostic information is also provided to the host system through the receive FIFO buffer. Loopback test connectors are not needed when running the user-mode diagnostics. A maintenance kit with service-mode diagnostics and loopback connectors is available from DIGITAL. A green LED indicates GO/NO-GO status for the device.

1.2.1.3 Preventing Data Loss -- The CXA16/CXB16 can be programmed for automatic XON and XOFF operation, to prevent the loss of data at high throughput. The reporting of received XON/XOFF characters to the software driver can be enabled or disabled.

1.2.2 Physical Description

The CXA16/CXB16 is based on a standard quad-height module 10.4 inches (26.4 cm) long and 8.4 inches (21.3 cm) wide. Figure 1-1 shows the layout. The spacing of the backplane slots into which the module connects is physically different from a standard Q-bus backplane, although they are electrically compatible. The serial-line interface is through two 36-way connectors, J1 and J2, mounted on the module handle. This handle forms part of the BA213/BA214 bulkhead. Connections to terminals and other peripheral devices from these connectors are made by two 36-wire cables (BC16D-25) and H3104 cable concentrators.

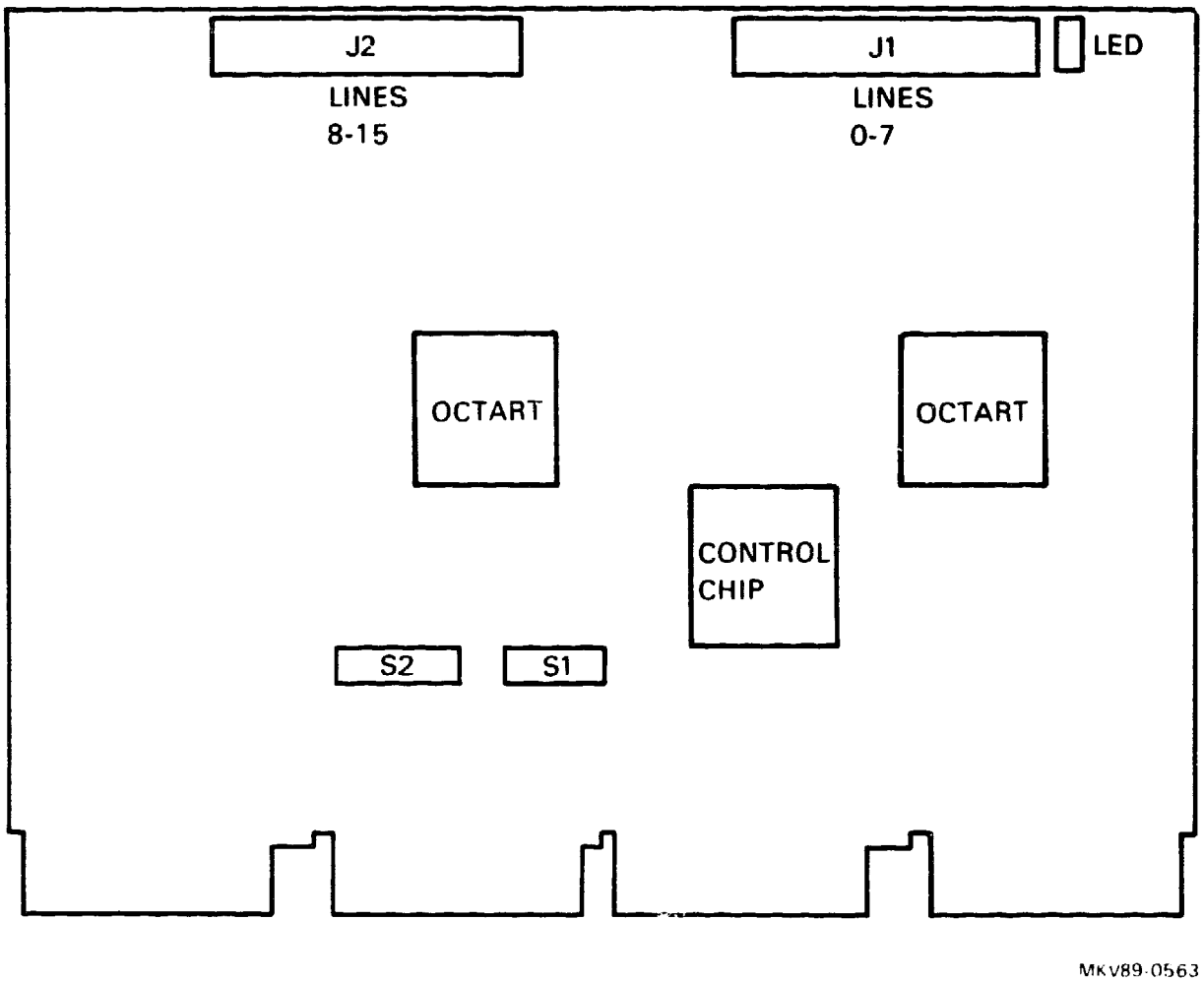


Figure 1-1 CXA16/CXB16 Module Layout

1.2.2.1 **On-Board Switchpacks** -- The CXA16/CXB16 has two on-board switchpacks to select the following device functions.

- Ten-position switchpack (S2)

  Switch position 1 selects DHU11 programming mode when open, or DHV11 mode when closed.

  Switch positions 2 to 10 select the device Q-bus address.

- Eight-position switchpack (S1)

  Switch position 1 enables the on-board oscillator. This is a manufacturing test switch, and must be **closed** for normal operation. The device will not function with the switch open.

  Switch position 2 selects the external loopback indicator for self-test. This is a manufacturing test switch, and must be **open** for normal operation.

  Switch positions 3 to 8 select the device-interrupt vector address.

1.2.3 Configurations

The CXA16/CXB16 can be used in many different system configurations. Figure 1-2 shows a typical application.

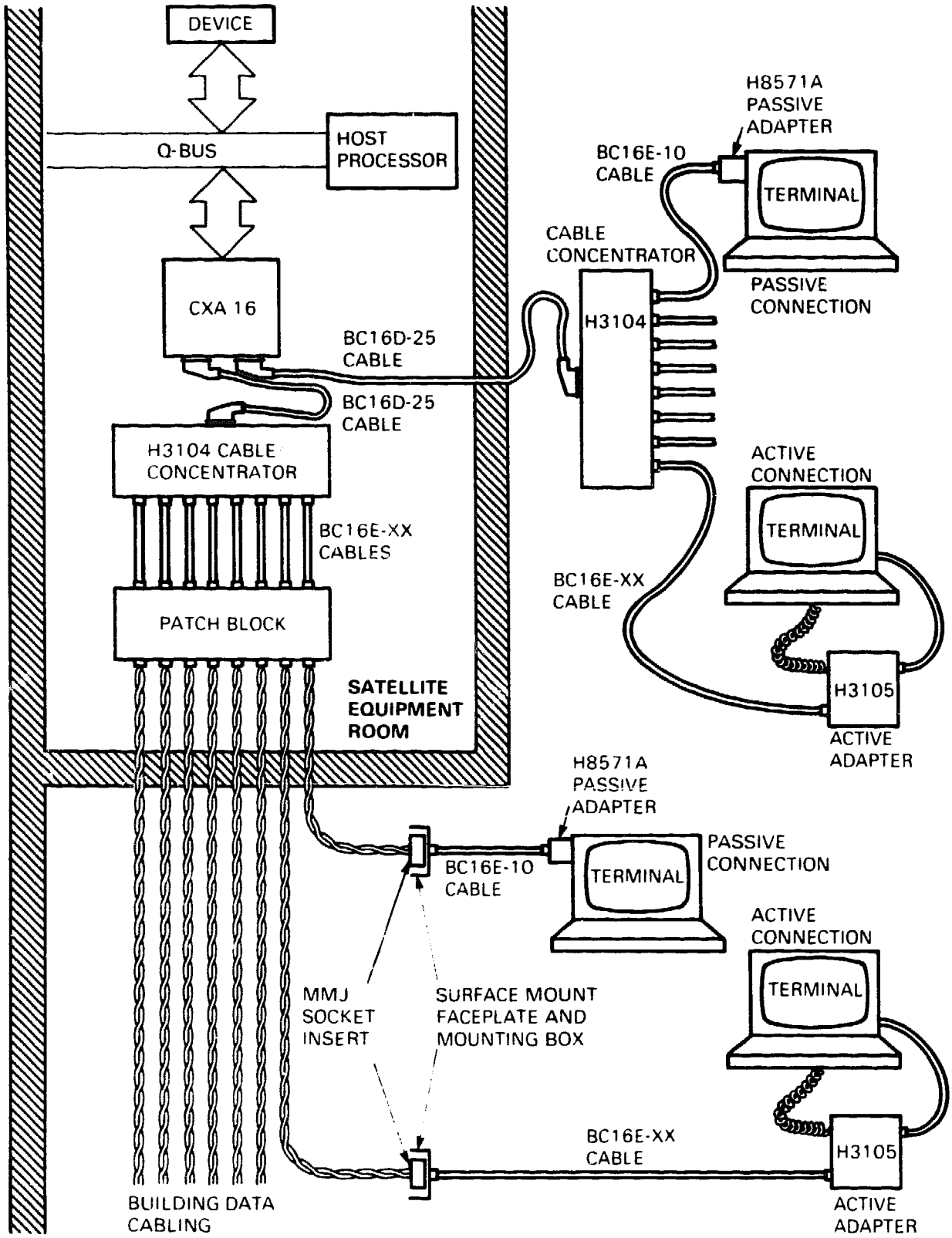


Figure 1-2 Example of CXA16 Configuration

1.2.4 Connections

The CXA16/CXB16 module is connected to the system backplane. Figure 1-3 shows the interconnections and test arrangements.

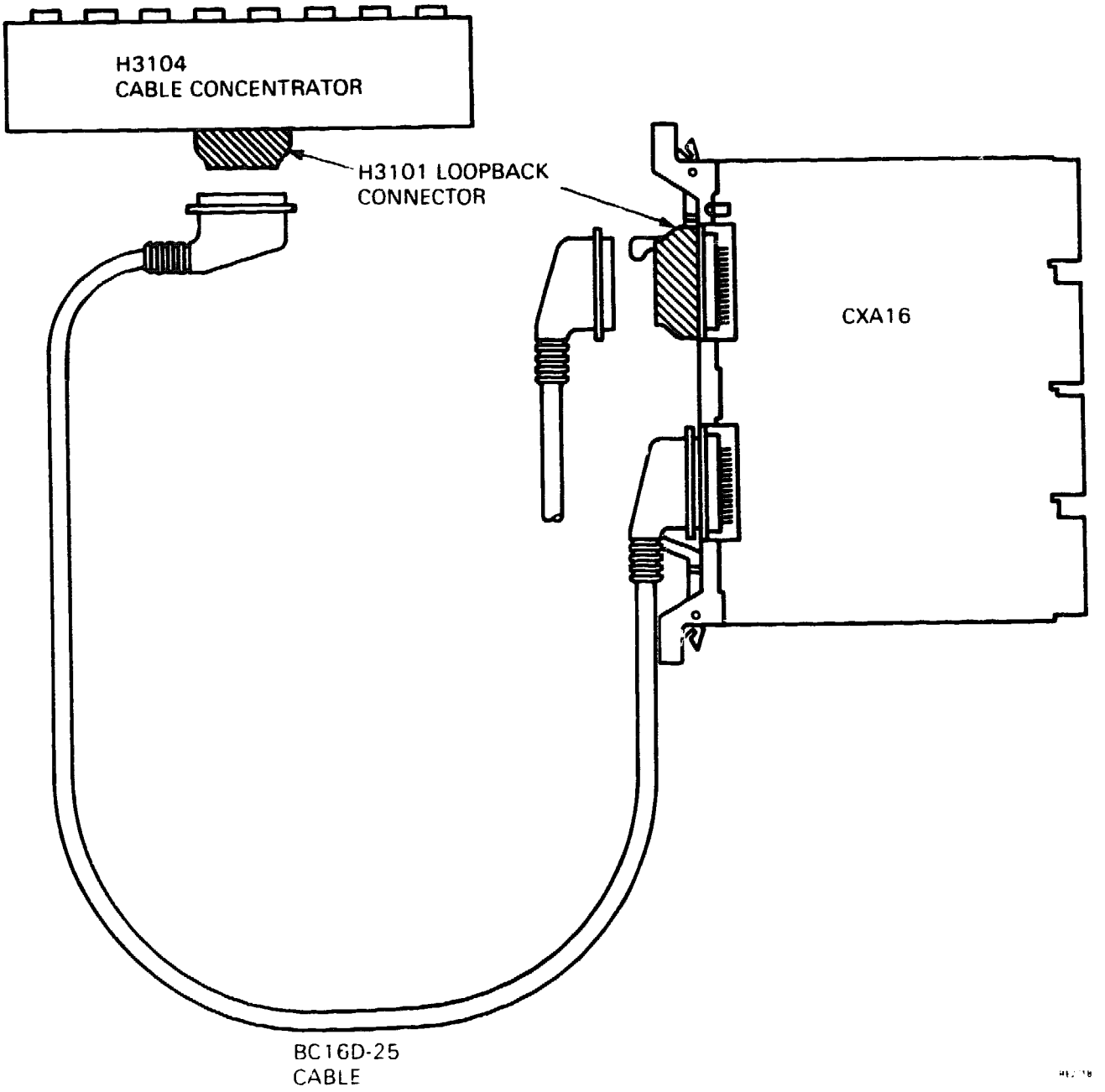


Figure 1-3 CXA16/CXB16 Connections

1.3 SPECIFICATION

1.3.1 Environmental Conditions

Environmental constraints for storage and operation of the CXA16/CXB16 are as follows.

- Storage temperature within the range -40 degrees C to 66 degrees C (-40 degrees F to 151 degrees F)
- Operating temperature within the range 5 degrees C to 60 degrees C (41 degrees F to 140 degrees F)
- Relative humidity within the range 10 percent to 95 percent, non-condensing, at a maximum wet-bulb temperature of 32 degrees C and a minimum dew point of 2 degrees C.

DIGITAL normally defines the the operating temperature range for a system as 5 degrees C to 50 degrees C (41 degrees F to 122 degrees F); the 10 degrees C difference between the upper limits quoted allows for the temperature gradient within the system box.

The maximum operating temperatures must be derated by 1.8 degrees C/1000 m (1 degree F/1000 ft) for operation at high-altitude sites.

1.3.2 Electrical Requirements

The CXA16/CXB16 needs the following electrical supplies (from the backplane).

- 5 V dc plus or minus 5 percent at 2.1 A maximum current, 1.4 A typical
- 12 V dc plus or minus 3 percent at 135 mA maximum, 110 mA typical (CXA16 only).

The CXA16 has an on-board switch-mode power supply, which generates a -10 V supply with the following characteristics:

- -10 V dc plus or minus 5 percent

1.3.2.1 Q-bus Loads -- The loads applied to the Q-bus are:

- 3.0 ac loads
- 1.5 dc loads.

The EIA-232-D/CCITT V.24/V.28 standard was originally designed to specify the connection between a local interface and a modem. It was not intended to be used for connecting to terminals over long distances. The maximum specified cable length is 50 ft (15 m). Shielded cable must be used in order to meet the requirements of FCC and VDE Radio Frequency Interference (RFI) regulations.

Although cable lengths greater than 50 ft can be used with reasonable success, cable capacitance, noise, and ground potential difference restrict the line speed as the distances increase. Consequently, the performance of long-distance communications to a terminal using EIA-232-D often does not meet today's requirements for terminal wiring.

DEC423 is a superset of the RS-423-A/CCITT V.10 standard. RS-423-A has a different grounding and signal return path arrangement from EIA-232-D, as well as using different line driver and receiver chips.

DEC423 uses line driver and receiver chips which have better filtering and tighter level tolerances than those specified by RS-423-A. In addition, DEC423 devices include transient suppressors for electrical overstress (EOS) and electrostatic discharge (ESD) protection. DEC423 devices may also be connected with unshielded cable.

The features provided by DEC423 devices provide reliable data communication over increased distances, typically 1000 ft (300 m) at 9600 baud. See Table 1-1 for maximum distance guidelines.

Table 1-1 Maximum Distance Guidelines for CXA16/CXB16

	Up to 4.8 Kb	9.6 Kb	19.2 Kb	38.4 Kb
DEC423 to DEC423	1000 ft 300 m	1000 ft 300 m	1000 ft 300 m	500 ft 150m
DEC423 to EIA-232-D	250 ft 75 m	200 ft 60 m	-	-
DEC422 to DEC422	4000 ft 1200 m	4000 ft 1200 m	4000 ft 1200 m	4000 ft 1200 m

DEC423 is for data-leads-only connections to terminal equipment, and is not suitable for connection to modems or other Wide Area Network equipment. The standard also specifies the use of a modified modular jack (MMJ) connector, instead of the much larger D-type connectors defined in RS-449.

DEC423 is signal compatible with the EIA-232-D standard when used for data-leads-only interconnection, in that interconnection between devices using the different standards is possible. However, the restrictions on speed and distance of EIA-232-D will still apply.

DEC423 should always be used in preference to EIA-232-D for direct terminal connection over extended distances.

NOTE

EIA-232-D terminals can be connected to the CXA16 using either of the following: an H8521 passive terminal adapter or an H3105 active terminal adapter. An H3105 active terminal adapter is necessary when using an EIA-232-D terminal with a DEC423 interface, if the longer cable lengths obtainable with DEC423 are required.

A terminal may be connected directly to the H3104 cable concentrator. The recommended cable is BC16E-xx, which has MMJ plugs at both ends, and is available in lengths up to 100 ft (30 m). The cable is also available without MMJ connectors in 1000 ft (300 m) reels, part number H8240. MMJs are available in packs of 50, part number H8220. There are many other DEC423 connection components available. For more information, contact your local sales office.

NOTE

DEC423 and EIA-232-D connections are intended for local communication. In general, communication devices can become non-operational or be damaged if the total cable length exceeds 1000 ft (300 m) for DEC423 devices, or 300 ft (100 m) for EIA-232-D devices. The cable should not be run outside the building, and the low-voltage data wiring must be separated from ac power wiring. The installation or site may require additional devices to correct problems in communication.

NOTE

Under ideal conditions, DEC423 devices can drive cables considerably longer than the 1000 ft maximum stated above. However, differences in ground potential, pick-up from mains ac power cabling, and risk of induced interference limits the maximum distance for reliable communications in most practical situations.

DEC422 also incorporates EOS/ESD protection and RS-422 compatible drivers and receivers. This makes it an ideal standard for use in high-noise environments.

Note that while connection between DEC422 and DEC423/EIA-232-D devices is feasible, it is not recommended or supported by DIGITAL for this product.

DEC422 supports cable lengths of up to 4000 ft (1200 m) at all supported speeds.

1.4.3 Line Receivers

The CXA16/CXB16 module uses octal serial-line receivers, which convert the input signals to TTL levels suitable for use by the OCTART. Signals are inverted by the receivers.

1.4.4 Line Transmitters

The CXA16 module uses octal serial-line transmitters, and the CXB16 uses quad line drivers. These convert the TTL level signals from the OCTART to DEC423 or DEC422 levels on the data lines. The signals are inverted by the transmitters.

Table 1-2 shows connections to the 36-pin connectors used on the CXA16/CXB16.

Table 1-2 Serial Line Connections for the 36-Pin Connector

1	Blu/Wht	Line 0 Transmit +	19	Wht/Blu	Line 0 Transmit -
2	Org/Wht	Line 0 Receive +	20	Wht/Org	Line 0 Receive -
3	Grn/Wht	Line 1 Transmit +	21	Wht/Grn	Line 1 Transmit -
4	Brn/Wht	Line 1 Receive +	22	Wht/Brn	Line 1 Receive -
5	Slt/Wht	Line 2 Transmit +	23	Wht/Slt	Line 2 Transmit -
6	Blu/Red	Line 2 Receive +	24	Red/Blu	Line 2 Receive -
7	Org/Red	Line 3 Transmit +	25	Red/Org	Line 3 Transmit -
8	Grn/Red	Line 3 Receive +	26	Red/Grn	Line 3 Receive -
9	Brn/Red	Line 4 Transmit +	27	Red/Brn	Line 4 Transmit -
10	Slt/Red	Line 4 Receive +	28	Red/Slt	Line 4 Receive -
11	Blu/Blk	Line 5 Transmit +	29	Blk/Blu	Line 5 Transmit -
12	Org/Blk	Line 5 Receive +	30	Blk/Org	Line 5 Receive -
13	Grn/Blk	Line 6 Transmit +	31	Blk/Grn	Line 6 Transmit -
14	Brn/Blk	Line 6 Receive +	32	Blk/Brn	Line 6 Receive -
15	Slt/Blk	Line 7 Transmit +	33	Blk/Slt	Line 7 Transmit -
16	Blu/Yel	Line 7 Receive +	34	Yel/Blu	Line 7 Receive -
17	Org/Yel	Spare	35	Yel/Org	Spare
18	Grn/Yel	Spare	36	Yel/Grn	Spare

1.5 FUNCTIONAL DESCRIPTION

1.5.1 General

The CXA16/CXB16 functional blocks are shown in Figure 1-4. Most of the functions are provided by three chips: one control chip and two OCTART chips.

Q-bus buffering uses six DC021 bidirectional buffers. Serial-line interface buffering uses two octal line receivers and two octal line transmitters for the CXA16, and two octal EIA receivers and four quad line transmitters for the CXB16.

A 2k x 8 static RAM chip provides the memory requirements. Switchpacks provide vector address, module address, and programming mode selection.

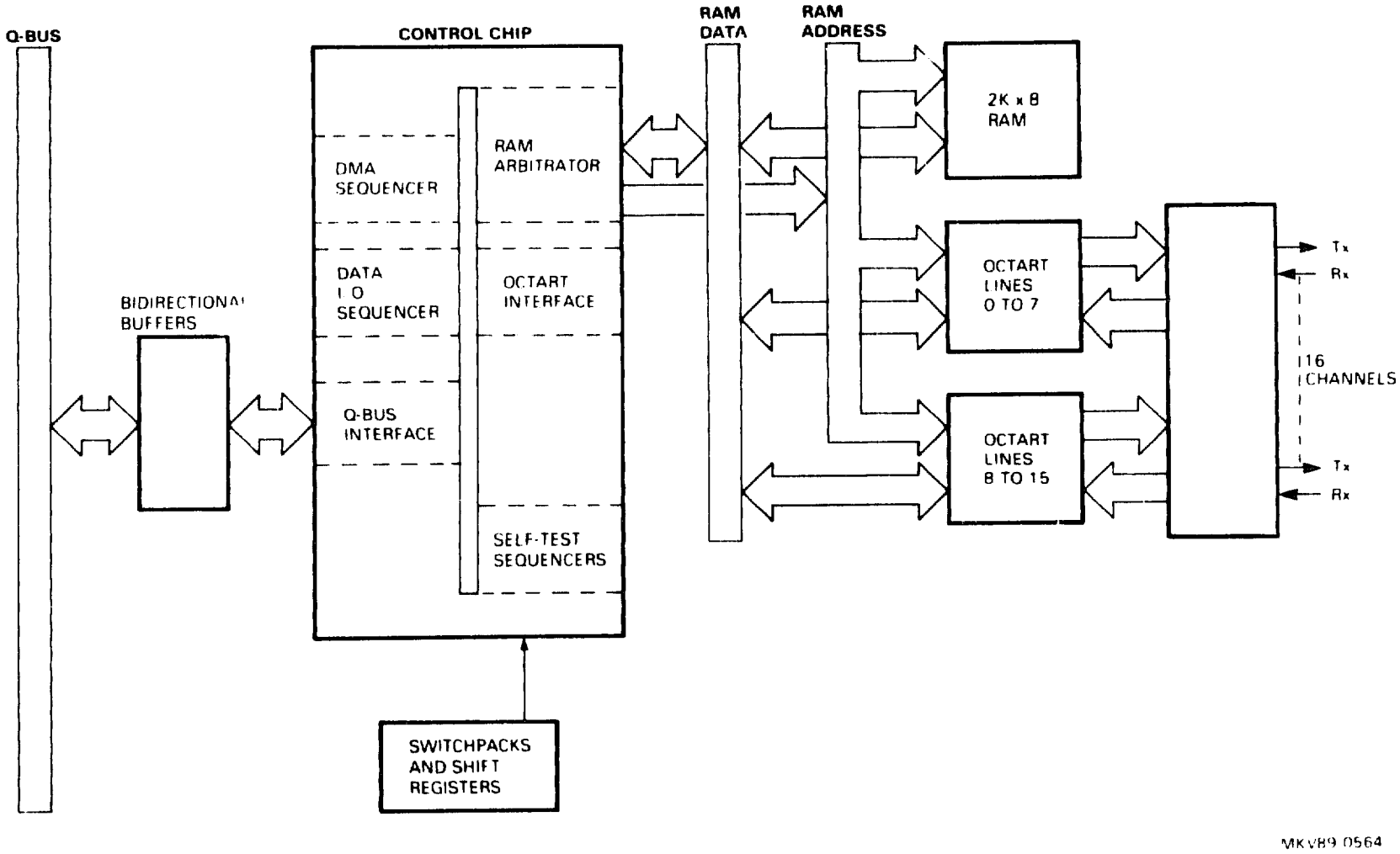


Figure 1-4 CXA16/CXB16 Functional Block Diagram

1.5.2 CXA16/CXB16 Main Functions

1.5.2.1 Transmission -- In the preferred programming mode (DHU11), characters may be either written directly to the transmit data FIFO (programmed transfer), or may be transferred from the host memory to the transmit data FIFO using DMA transfers.

In DHV11 programming mode, only single characters can be transmitted using programmed transfers. Characters can also be transferred by DMA as in DHU11 programming mode.

1.5.2.2 Reception -- Received characters are deserialized by the OCTART and transferred to a four-character area in the RAM (one such area per line) by the control chip OCTART interface, following an interrupt from the OCTART. The OCTART interface later removes characters from the bottom of this 4-character FIFO, and places it in the 256 x 16 receive FIFO, which can be read by the host.

1.5.3 Control Chip

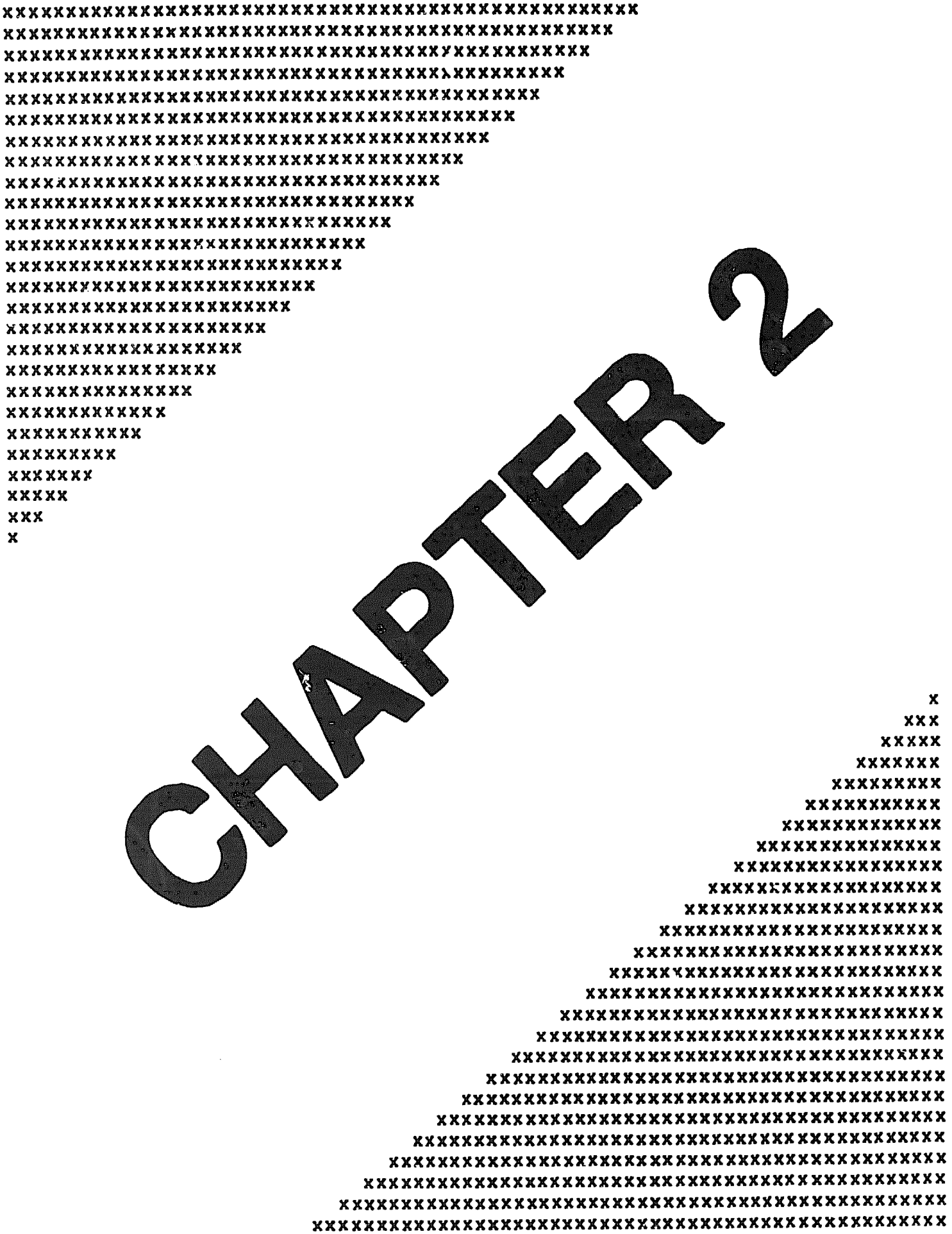
The control chip contains the following functional blocks.

- Q-bus Interface -- Matches addresses, generates vector addresses, and handles interrupts. It also interfaces the Q-bus signals to other functional blocks.
- DIO Control -- Controls host access to device registers.
- OCTART Interface -- Transfers data between the OCTARTs and RAM, and handles flow control. It also controls the operation of the OCTART.
- Self-Test/Power-Up Sequencer -- This section powers up the module to a fixed set of initial conditions, such as 9600 baud rate on all lines; it also handles self-test.
- DMA Sequencer -- Initiates and manages all DMA data transfers to the module.
- RAM Arbitrator -- Provides RAM and OCTART bus access to the various sequencers.

1.5.4 OCTART Chip

This chip contains eight UARTs, which perform parallel-to-serial and serial-to-parallel data conversions. It interfaces with the control chip through eight registers. Four are read-only and four are write-only. An index register is used to access individual lines. The OCTART chip shares the RAM bus with the control chip, the RAM itself, and a second OCTART chip. The OCTART chip also includes:

- Receive and transmit control blocks
- Interrupt logic for interfacing with the control chip
- A 16-output baud-rate generator
- All necessary line-parameter registers
- Diagnostic loopback logic
- Modem status multiplexers.


CHAPTER 2

CHAPTER 2
INSTALLATION

2.1 SCOPE

This chapter describes the preparation and installation of the CXA16/CXB16 as add-on options. If the option was ordered as part of a system, it will be already installed, and you should refer to the system unpacking instructions. This chapter contains the following sections.

- Unpacking
- Preparation
- Installation
- Testing.

The procedures described in this section are designed for use only by qualified service engineers.

2.2 UNPACKING AND INSPECTION

There are two options available, the CXA16-AA and the CXB16-AA. The contents of each option are as follows.

Item	Part number	Quantity CXA16-AA	Quantity CXB16-AA
CXA16 module	M3118-YA	1	
CXB16 module	M3118-YB		1
Cable concentrator	H3104	2	2
Cable	BC16D-25	2	2
Loopback connector	H3101	1	1
User Guide	EK-CAB16-UG	1	1
Labels		1	1

Undo each package and examine the contents for physical damage. Check that the contents of each package are complete. Report any damaged or missing items to the shipper and to the local DIGITAL office. Do not dispose of the packing material until the unit has been installed and is operational.

2.3 PREPARING THE CXA16/CXB16 MODULE

Three parameters must be defined before installation.

- Q-bus base address
- Q-bus interrupt vector
- DHV11 or DHU11 programming mode.

These are selected using switchpacks on the CXA16/CXB16 module.

2.3.1 Address and Vector Assignment

Switchpack S1 and Switchpack S2 determine the device address and vector for the CXA16/CXB16 module respectively. The CXA16/CXB16's factory settings are only correct if no other floating address or vector options are installed in the microsystem. If you have other options, or more than one CXA16/CXB16, check your host system manuals; they may give recommended addresses and vector settings for the different options. If this information is not available, the full assignment rules must be applied; these are given in Appendix B.

NOTE

Some systems have additional restrictions on module addresses and on slot locations.

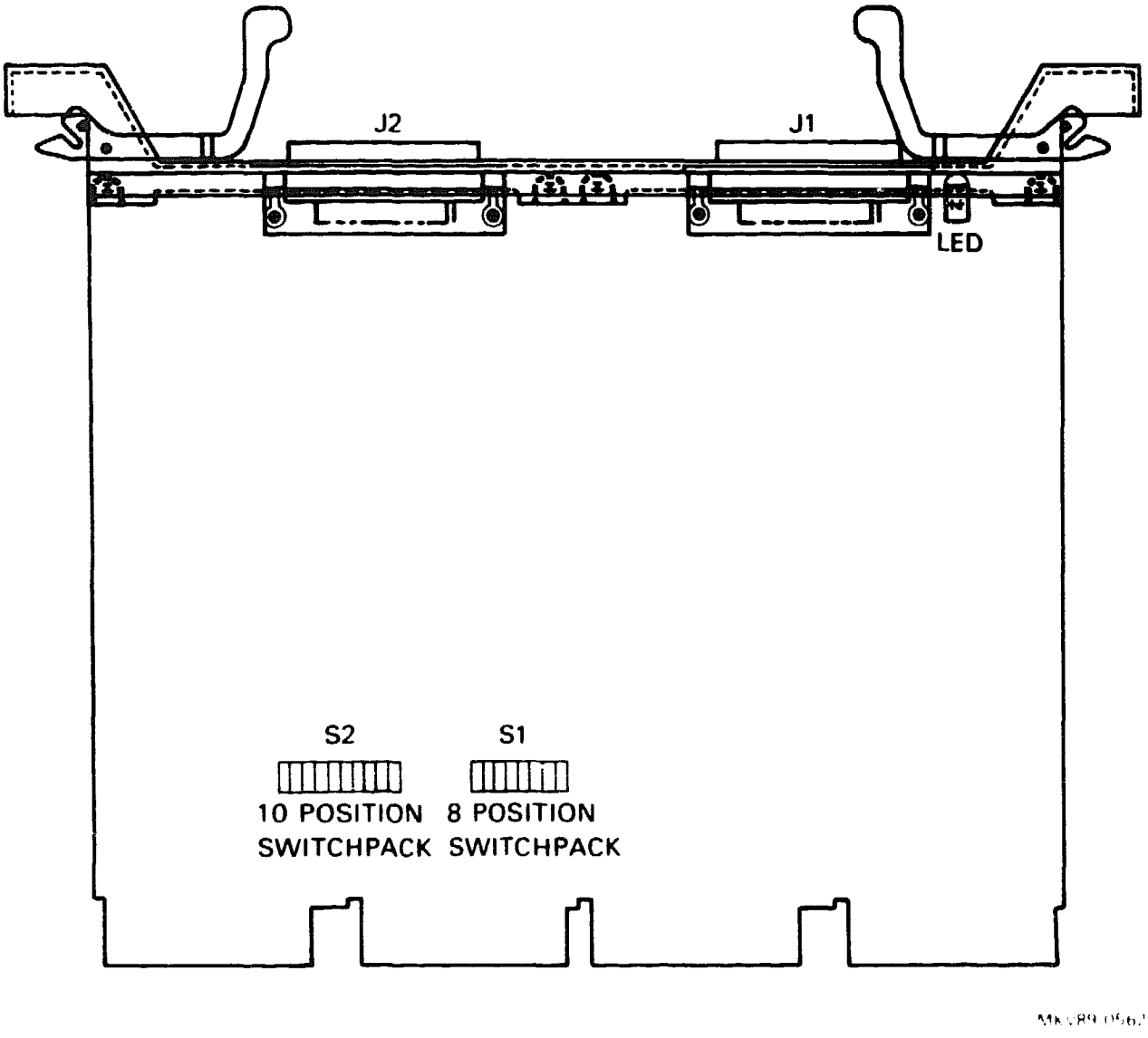


Figure 2-1 Location of the Switchpacks

2.3.2 DHV11 or DHU11 Programming Mode Selection

The CXA16/CXB16 offers two separate programming modes, DHV11 and DHU11. The mode selected depends on the device driver used in your system. The mode is determined by switch position 1 of the 10-position switchpack S2 (see Figures 2-1 and 2-2).

NOTE

DHU11 programming mode generally gives better performance because of reduced CPU overhead transferring characters to and from the device. The Software Product Description states whether the operating system supports DHU11 programming mode. DHU11 programming mode is the preferred mode of operation.

2.3.3 Setting the Address Switches

The device address for the CXA16/CXB16 is set on the 10-position switchpack S2. The location of this switchpack is shown in Figure 2-1. Figure 2-2 shows how to set the device address on the switchpack. The example shown is for the address of 17760440 (octal). (Switch 1 on the switchpack is for DHV11/DHU11 programming-mode selection.)

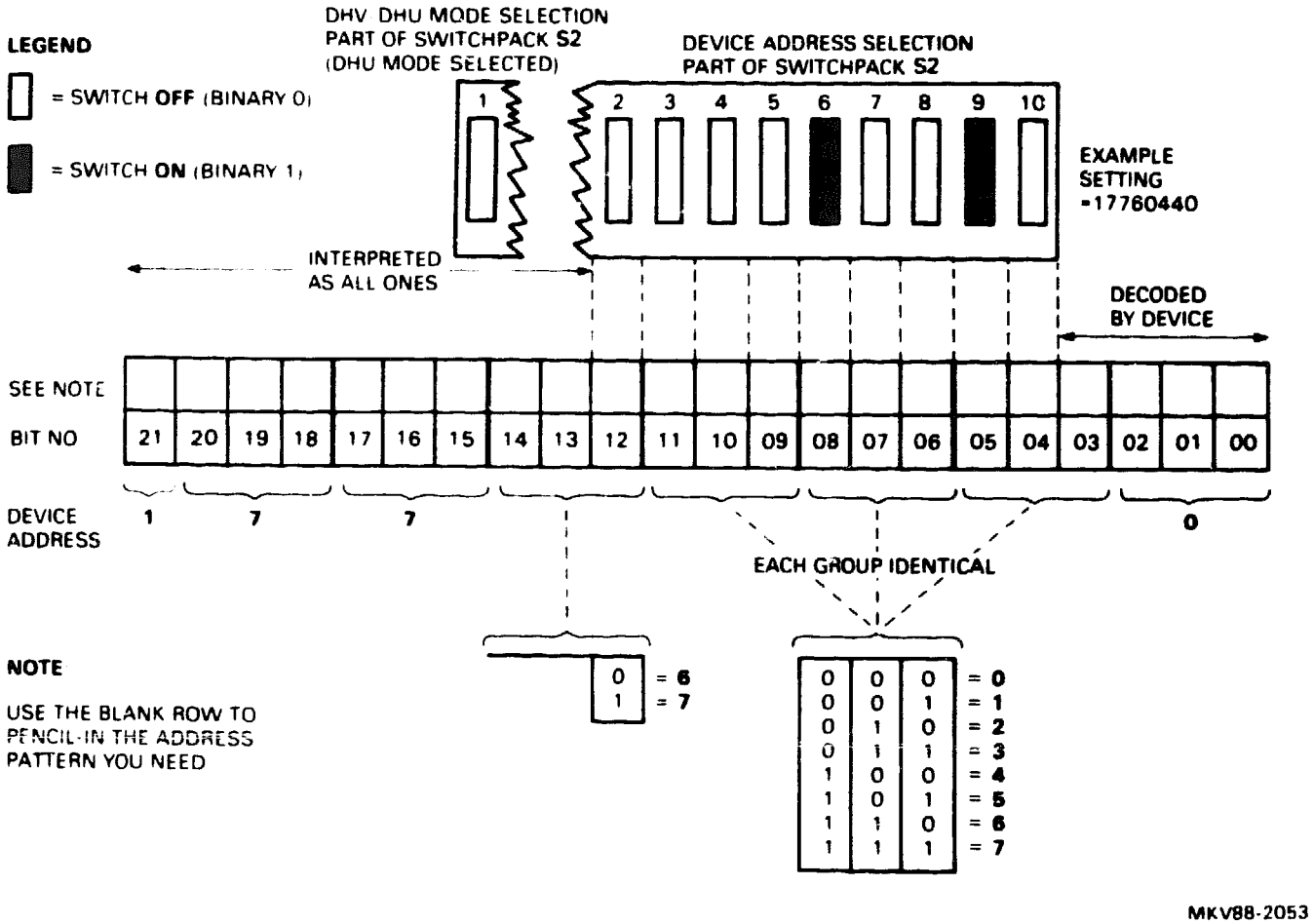


Figure 2-2 Setting the Device Address

2.3.4 Setting the Vector Switches

The six high-order bits of the interrupt vector address are set on the eight-position switchpack S1. Figure 2-1 shows the location of this switchpack. Figure 2-3 shows an example of these switches set to 300 (octal). (Switch positions 1 and 2 are used during manufacture, switch position 1 must be **closed**, and switch position 2 must be **open** for correct operation of the CXA16/CXB16.)

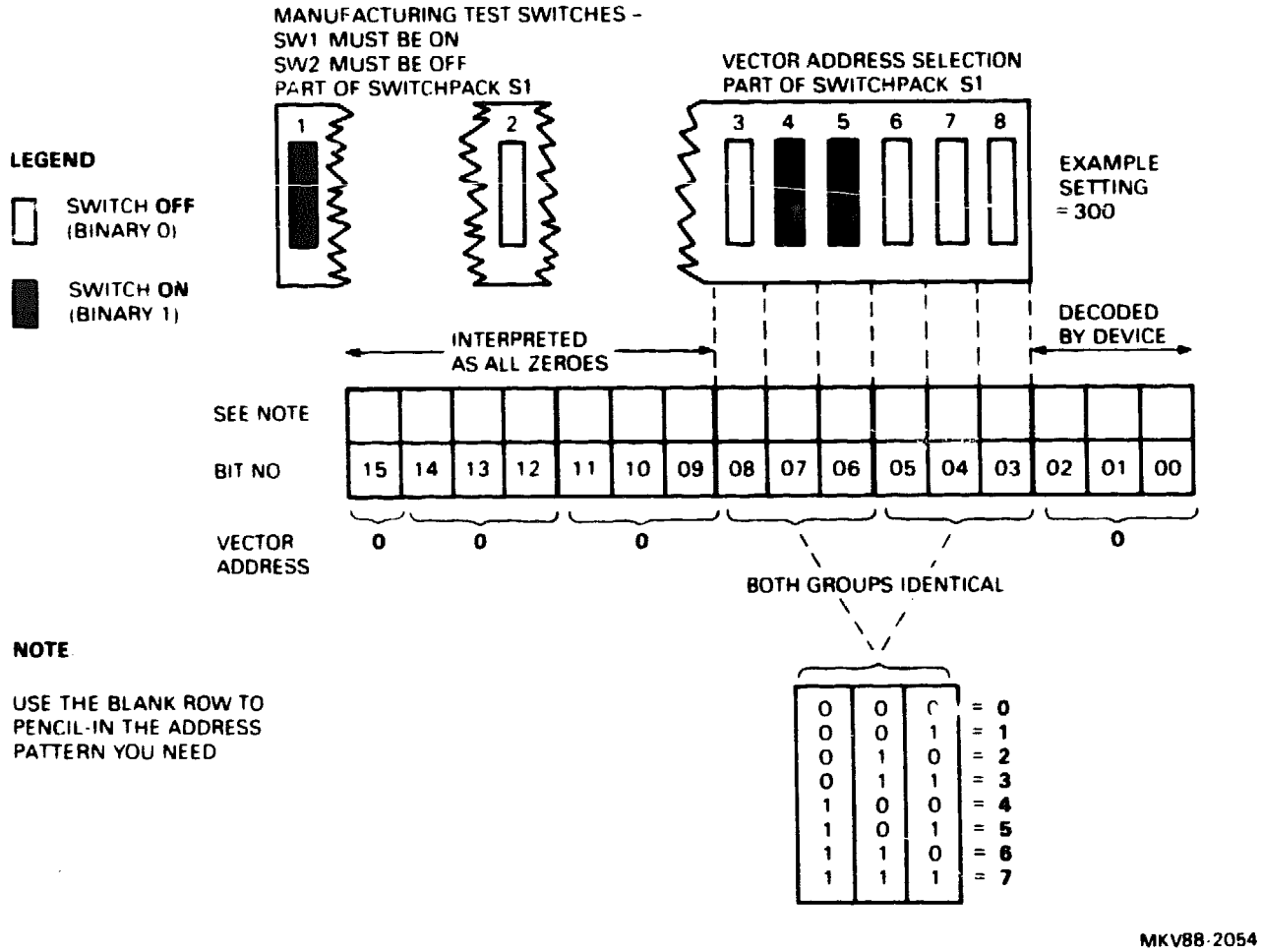


Figure 2-3 Setting the Vector Address

2.4 BUS CONTINUITY

Bus grant continuity jumper cards are used in vacant backplane slots to provide bus continuity.

2.4.1 Bus Grant Continuity Jumpers

The CXA16/CXB16 uses the Q/CD backplane, Q-bus on A and B connectors, user-defined signals on C and D.

In Q/CD backplanes, bus grant signals pass through each installed module through the A connectors of each bus slot.

Lines AM2 and AN2 (BIAK), and AR2 and AS2 BDMG), carry the bus grant signals. Figure 2-4 shows the bus grant routing.

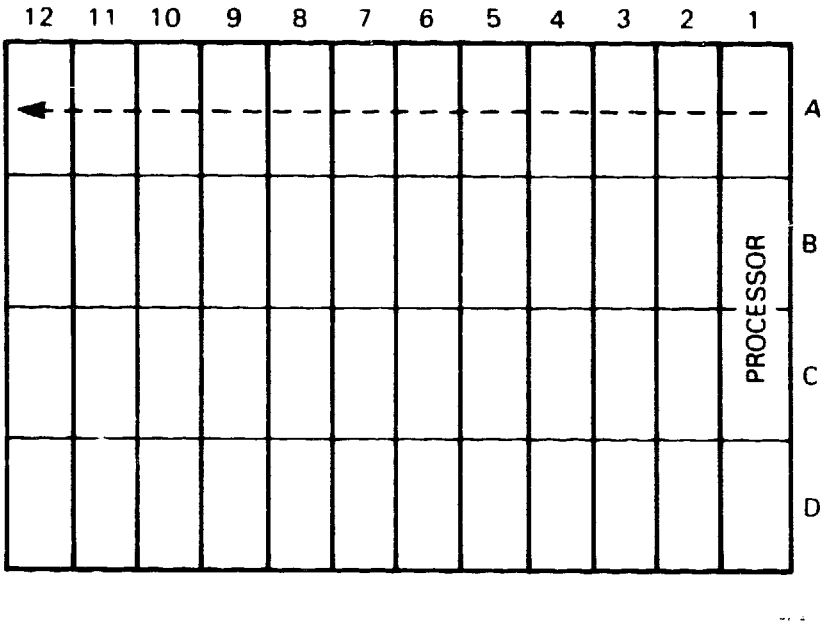


Figure 2-4 Bus Grant Continuity Flow

2.5 PRIORITY SELECTION

The CXA16/CXB16 uses the BIRQ4 line to request interrupt service. It does not monitor any of the higher-level interrupt request lines. Because of this, both the interrupt-request and DMA (non-processor request) priorities of the CXA16/CXB16 are selected by the position of the CXA16/CXB16 on the bus. Devices closest to the processor module have the highest priority.

The bus (backplane) position may be a compromise between DMA and interrupt priority requirements. As a general rule, consider DMA request priorities first, then interrupt (bus) requests.

2.5.1 Recommendations

In general the CXA16/CXB16 bus position is not critical. However, it is recommended that you place the module after any mass-storage interfaces and high-speed synchronous communications options; these are more sensitive to bus position.

The following list shows the recommended module sequence:

1. CPU
2. Memory modules
3. Synchronous communication modules - no silo
4. General-purpose I/O ports
5. Line printer interface
6. Asynchronous communications modules - no silo
7. Asynchronous communications modules - silo
8. Synchronous communication modules - DMA
9. Communications module - smart DMA
10. Asynchronous communications modules - silo/DMA (e.g. CXA16/CXB16).

2.5.2 DMA Request Priority

DMA request priority is usually assigned according to throughput. Faster devices (higher throughput) usually have priority over slower DMA devices; for example, disk has priority over tape, which itself has priority over communications devices. This is because fast devices usually reach overrun or underrun conditions sooner than slower ones.

2.6 CXA16/CXB16 INSTALLATION

This section gives a step-by-step guide to installing the option. Figure 2-5 shows how the parts of the option connect together.

1. Before you begin the installation, check that the system is operating correctly by running appropriate diagnostics.

2. Turn the system power off.

WARNING

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

ATTENTION

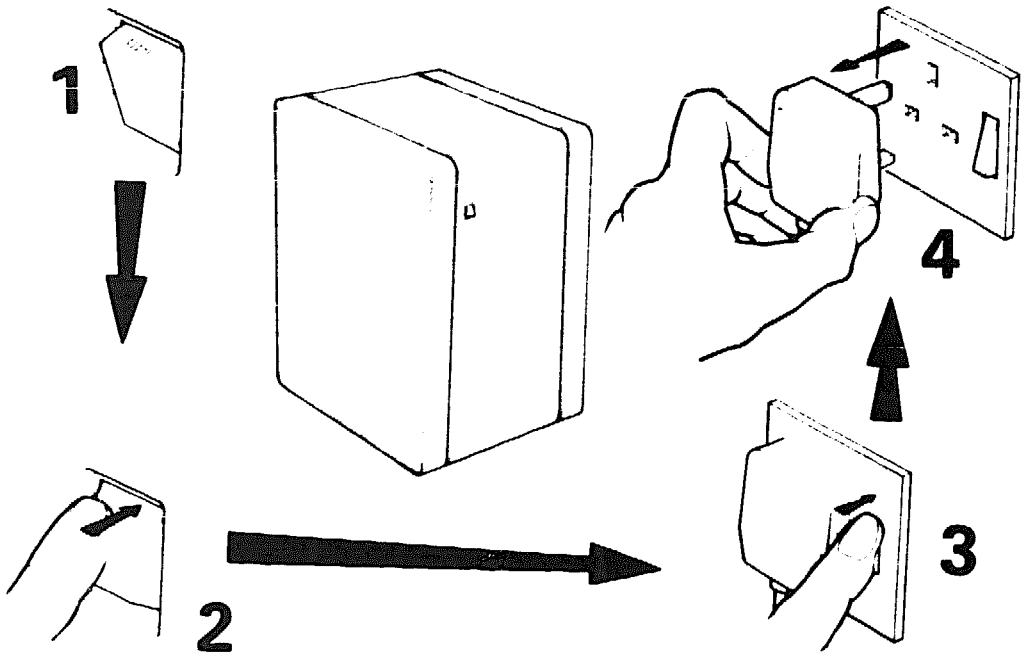
Avant d'effectuer l'une des procédures de ce chapitre, mettez le système hors tension et débranchez le cordon d'alimentation.

VORSICHT!

Schalten Sie das System ab, und ziehen Sie das Netzkabel, bevor Sie die in diesem Kapitel beschriebenen Answeisungen ausführen.

ATENCION

Apague el sistema y desconecte el cable principal de alimentación antes de realizar ningún procedimiento de este capítulo.



RE/848

3. Remove the module blanking plate at the chosen position in the system box and the grant card if fitted.

NOTE

Make sure that bus grant continuity is maintained from the CPU to the last module on the bus.

4. Insert the module into the backplane.

NOTES

1. Be careful not to snag module components on the card guides or adjacent modules.

2. Ensure that you are wearing an antistatic wriststrap, part number 29-11762-00.

5. Turn the power on. After two seconds check that the self-test LED is on; this indicates a successful self-test*. Correct any problems before proceeding with the installation.

6. Run the user-mode diagnostics* to test the module; no test connectors are required.

7. Connect the BC16D-25 cables to J1 and J2 on the module bulkhead.

8. Connect the other ends of each BC16D cable to an H3104 cable concentrator.

* The self-test and diagnostics used to test the installation of the CXA16/CXB16 are briefly described in Section 2.7, and in more detail in Chapter 4.

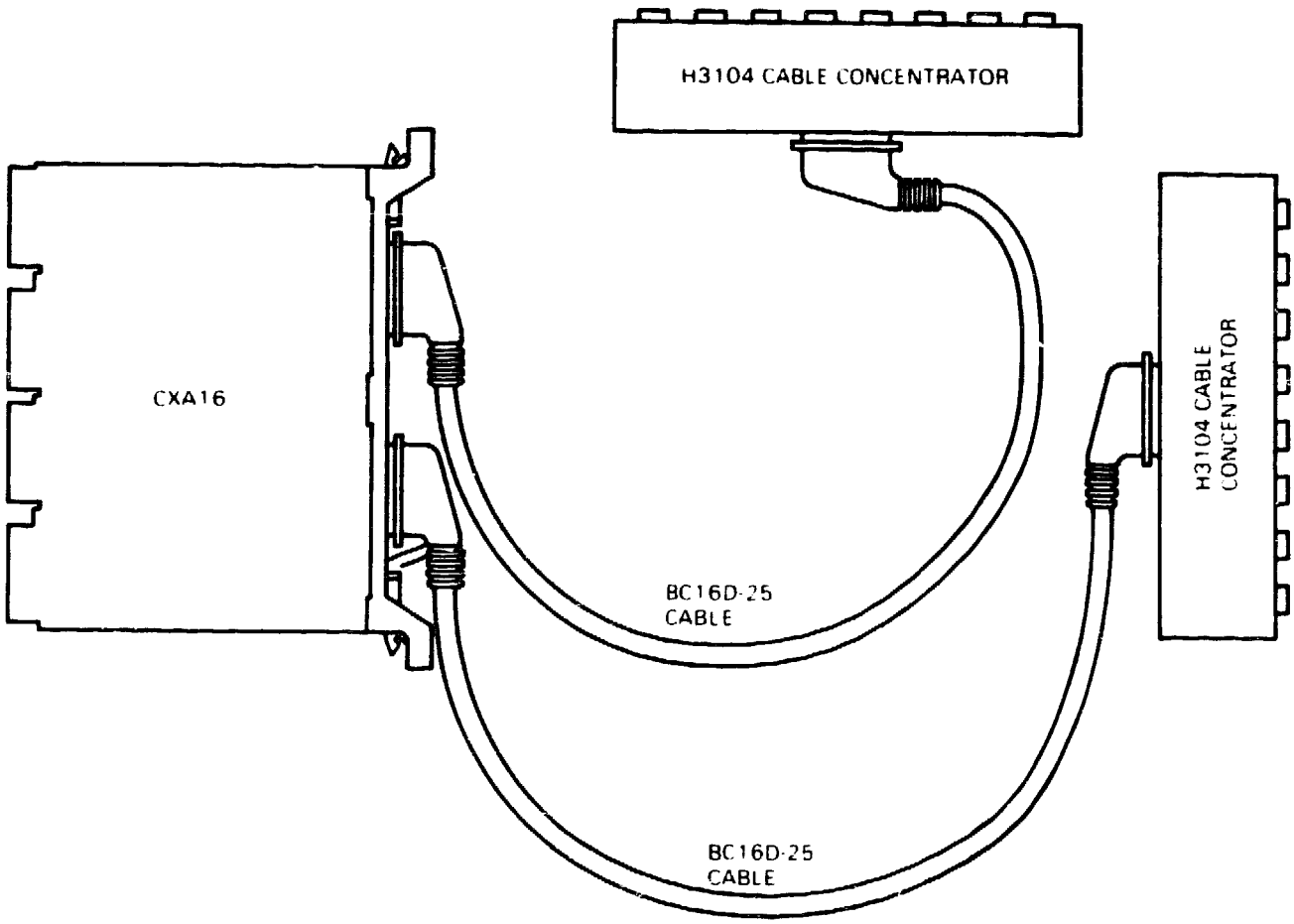


Figure 2-5 CXA16/CXB16 Installation

2.7 INSTALLATION TESTING

This section details the diagnostics used to test the option during and after installation. The diagnostics are also used to test other Q-bus modules in the same family, for example, DHV11. The diagnostics will automatically 'size' the option to determine which one is being tested. The CXA16/CXB16 can be tested in either of its two programming modes.

Both PDP-11 and MicroVAX II diagnostics are described. After successful completion of the appropriate system test, the CXA16/CXB16 may be connected to external equipment. Further information on the diagnostics is given in Chapter 4.

2.7.1 Installation Tests on MicroPDP-11 Systems

To verify that the MicroPDP-11 system and the CXA16/CXB16 module are functioning correctly:

1. Check that the green self-test LED on the CXA16/CXB16 module is on.

2. Boot the MicroPDP-11 Customer Diagnostic media. Refer to your MicroPDP-11 System Manual for further information.

3. Type 'I' at the main menu to allow the diagnostics to identify the new module and add it to the configuration file.

NOTE

Look at the list of devices displayed, and make sure that the new module is included. If it is not included, repeat the installation sequence, and make sure that the module switches have been set correctly.

4. Type 'T' at the main menu to run the system tests. These should complete without error; if an error occurs, call DIGITAL Field Service.

A MicroPDP-11 Maintenance Kit is available, which allows you to run individual diagnostic programs under the XXDP+ diagnostic monitor, and to configure and run DECX11 system test programs. The XXDP+ functional diagnostic is VHQA**.BIN, and the DECX11 module is XDHV**.OBJ.

2.7.2 Testing in MicroVAX II Systems

To verify that the MicroVAX II system and the CXA16/CXB16 module are functioning correctly:

1. Check that the green self-test LED on the CXA16/CXB16 module is on.

2. Boot the MicroVAX Maintenance System media. Refer to your MicroVAX II System Manual for further information.

3. Type '2' at the main menu to show the system configuration and devices.

NOTE

Look at the list of devices displayed, and make sure that the new module is included. If it is not included, repeat the installation sequence, and make sure that the module switches have been set correctly.

4. Type '1' at the main menu to run the system tests. These should complete without error; if an error occurs, call DIGITAL Field Service.

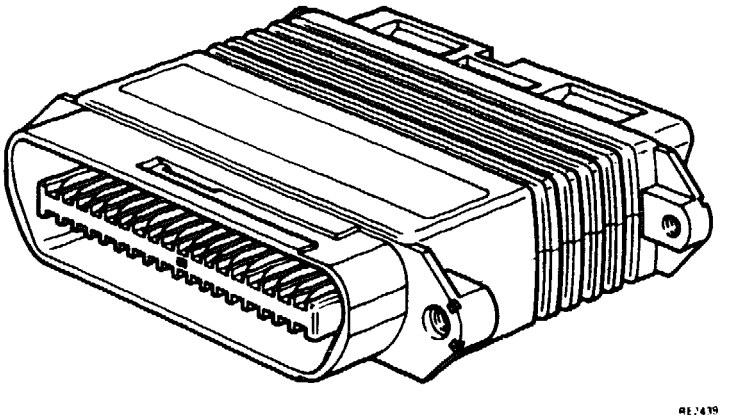
2.8 LOOPBACK TEST CONNECTORS

Two loopback connectors are available for the CXA16/CXB16, the H3101 and the H3103; only the H3101 is shipped with the option.

2.8.1 H3101 Loopback Connector

The H3101 loopback connector is two loopback connectors in one package, and consists of a female 36-way loopback connector and a male 36-way loopback connector. It can be inserted into the cabling at the module bulkhead, or at the cable concentrator. To test the cables, type characters at the keyboard and make sure that they are echoed to the screen (refer to Chapter 4).

Figure 2-6 shows the wiring of the H3101 loopback connector.



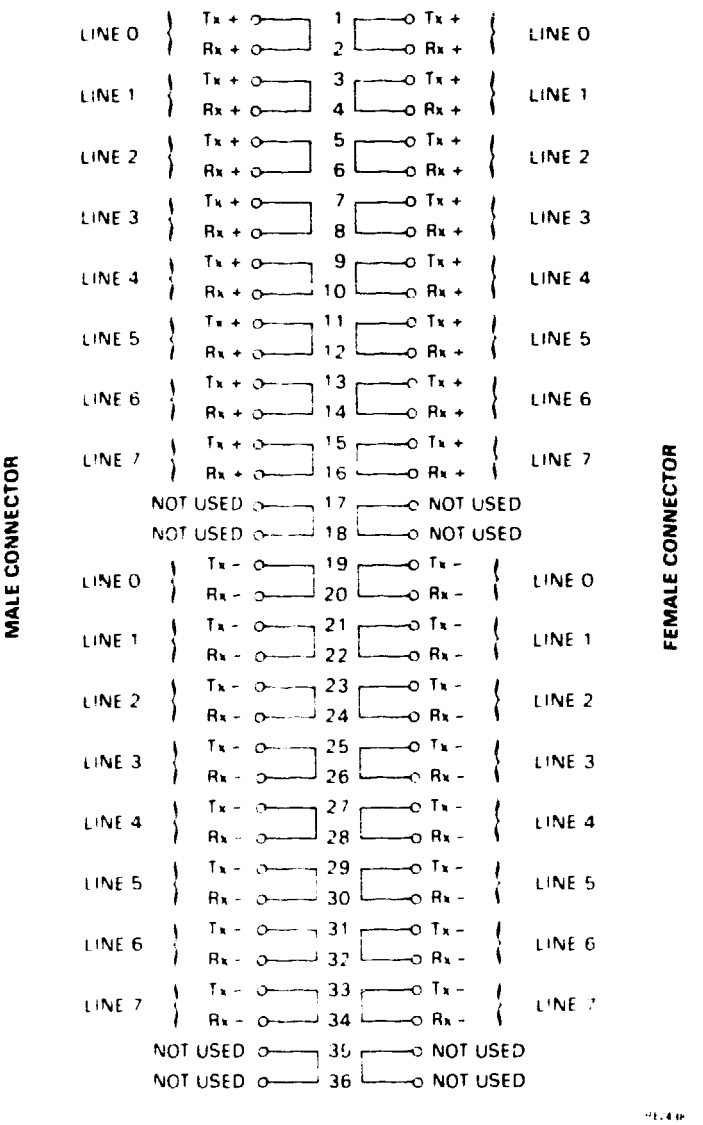
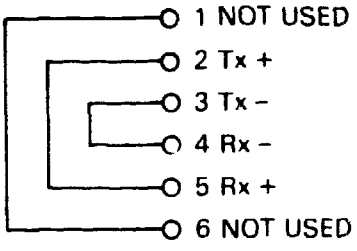


Figure 2-6 H3101 Loopback Connector

2.8.2 H3103 Loopback Connector

The H3103 loopback connector is used during maintenance diagnostic testing to test each line from the CXA16/CXB16 at the output from the H3104 cable concentrator. This connector is only available as part of the maintenance kit.

Figure 2-7 shows the wiring of the H3103 loopback connector.



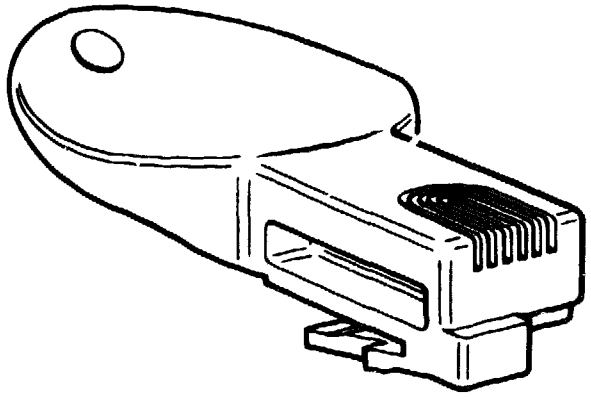
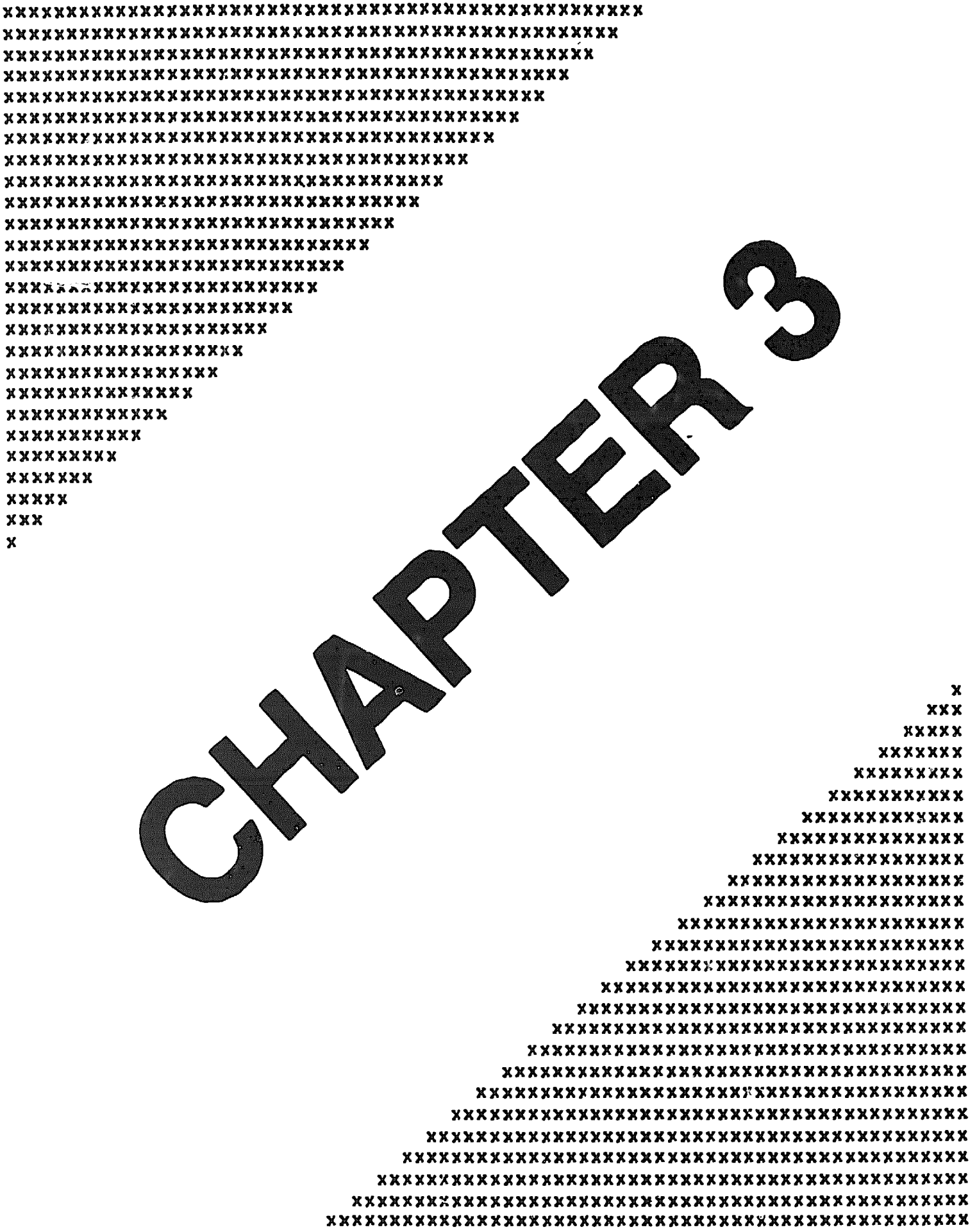


Figure 2-7 Line Loopback Test Connector


CHAPTER 3

CHAPTER 3
PROGRAMMING

3.1 SCOPE

This chapter describes the program control registers, and how they are used to control and monitor the CXA16/CXB16. The chapter covers:

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

Some programming examples are also included.

NOTE

DHU11 programming mode is the preferred mode of operation for the CXA16/CXB16. The development of user drivers that use the CXA16/CXB16 in DHV11 programming mode is not recommended.

3.2 REGISTERS

The host system controls and monitors the CXA16/CXB16 module through several Q-bus-addressable registers.

Command words or bytes written to the registers are interpreted and executed by the sequencers. Status reports and data are also transferred through the registers.

3.2.1 Register Access

CXA16/CXB16 registers occupy 8 words (16 bytes) of Q-bus memory-mapped I/O space. Some of these registers perform several functions, selected through bits in the CSR.

The base physical address of the eight CXA16/CXB16 registers is selected by switches on the module. The address selected must be in the peripheral I/O space.

Tables 3-1 and 3-2 list the CXA16/CXB16 registers and their addresses (in DHV11 mode and DHU11 mode). The suffix (I) means that there are eight of these registers, one for each channel. When an (I) register is accessed, the address given in the table is indexed by the contents of CSR<3:0> to select the register for the appropriate channel.

The term 'base' means the lowest I/O address on the module; that is to say, when the four low-order address bits = 0.

NOTE

Before indexed (I) registers are accessed, the channel number must be written to the CSR.

Table 3-1 CXA16/CXB16 Registers in DHV11 Mode

Register		Address (Octal)	Type
Control and Status Register	(CSR)	Base	Read/Write
Receive Buffer	(RBUF)	Base+2	Read Only
Transmit Character	(TXCHAR)	Base+2 (I)	Write Only
Line-Parameter Register	(LPR)	Base+4 (I)	Read/Write
Line Status	(STAT)	Base+6 (I)	Read Only
Line Control	(LNCTRL)	Base+10(I)	Read/Write
Transmit Buffer Address 1	(TBUFFAD1)	Base+12(I)	Read/Write
Transmit Buffer Address 2	(TBUFFAD2)	Base+14(I)	Read/Write
Transmit Buffer Count	(TBUFFCT)	Base+16(I)	Read/Write

NOTE

In DHV11 mode it is possible to write to the line-status register. However, the host should not write to this register.

Table 3-2 CXA16/CXB16 Registers when in DHU11 Mode

Register		Address (Octal)	Type
Control and Status Register	(CSR)	Base	Read/Write
Receive Buffer	(RBUFF)	Base+2	Read
Receive Timer*	(RXTIMER)	Base+2	Write(byte)
Line-Parameter Register	(LPR)	Base+4 (I)	Read/Write
FIFO Data	(FIFODATA)	Base+6 (I)	Write
FIFO Size	(FIFOSIZE)	Base+6 (I)	Read(byte)
Line Status	(STAT)	Base+7 (I)	Read(byte)
Line Control	(LNCTRL)	Base+10(I)	Read/Write
Transmit Buffer Address 1	(TBUFFAD1)	Base+12(I)	Read/Write
Transmit Buffer Address 2	(TBUFFAD2)	Base+14(I)	Read/Write
Transmit Buffer Count	(TBUFFCT)	Base+16(I)	Read/Write

* Only accessible when CSR<3:0>=0000.

There are 16 line-parameter registers, only 1 of which is accessed at any one time. The register which is accessed is associated with the line selected using CSR<3:0>. However, read-modify-write instructions may be used on all registers except CSR and RBUF.

For example, to read the line-parameter register of channel 3, the following I/O commands would be executed:

```
MOVB  #CHAN,@#BASE     ;WRITE CHANNEL NUMBER (SEE BELOW) TO  CSR
MOVB  @#BASE+4,R0      ;READ THE LINE PARAMETER REGISTER
```

In the above example, CHAN = 0er0011(binary)

Where

- e = the RXIE bit of the CSR
- r = the MASTER.RESET bit (which would be 0)
- 0011 = channel number 3

NOTE

Not all register bits are used. In a write action, all unused bits must be written as 0s. In a read action, unused bits are undefined.

3.2.2 Register Bit Definitions

Registers which are modified by reset sequences are coded as shown in Figure 3-1.

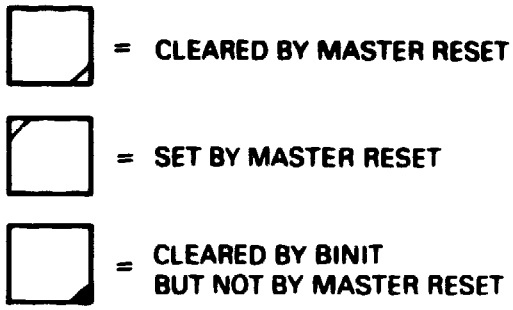
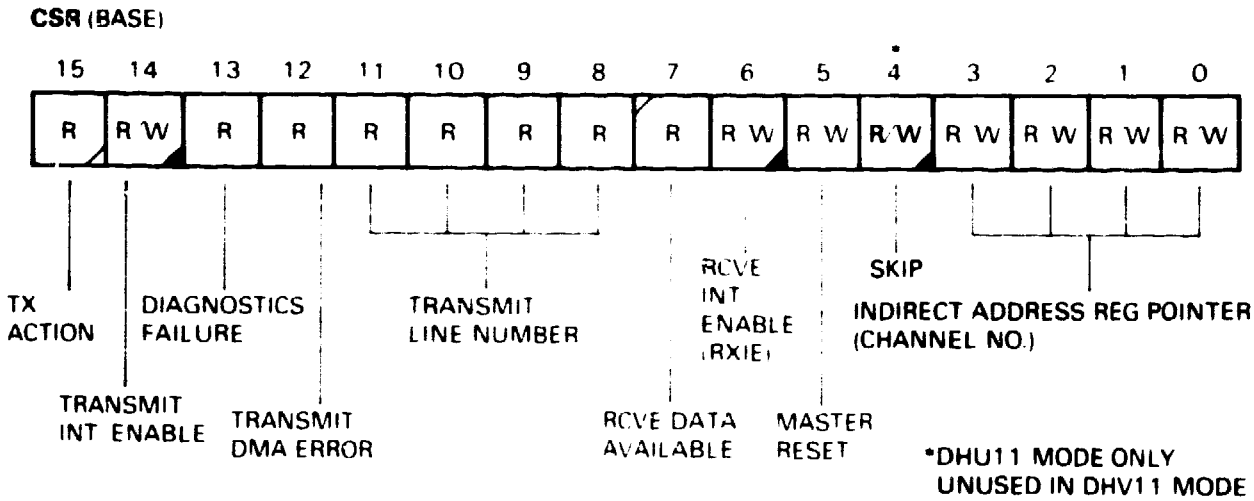


Figure 3-1 Register Coding

3.2.2.1 Control and Status Register (CSR) --



Bit	Name	Description
15	TX.ACTION (Transmitter Action) (R)	This bit is set by the CXA16/CXB16 when: 1. The last character of a DMA buffer has left the OCTART. 2. DMA transfer has been aborted. 3. DMA transfer has been terminated by CXA16/CXB16 because non-existent memory was addressed, or because of a memory parity error. 4. In DHV11 mode when a single-character programmed output has been accepted, that is, a character taken from TX.BUFF. 5. In DHU11 mode, following a programmed data transfer, the module has emptied a transmit FIFO. The bit is cleared if the host reads the CSR after the TX.ACTION FIFO has become empty. To avoid losing TX.ACTION reports, the host must not let more than 16 reports accumulate. It is advisable to read the CSR until TX.ACTION becomes clear. **NOTE** **TX.ACTION reports may be lost if the upper byte of the CSR is discarded following a read of the CSR.**
14	TXIE (Transmit Interrupt Enable) (R/W)	When set, this bit allows the CXA16/CXB16 to interrupt the host when CSR<15> (TX.ACTION) becomes set. The bit is cleared by BINIT, but not by MASTER.RESET.

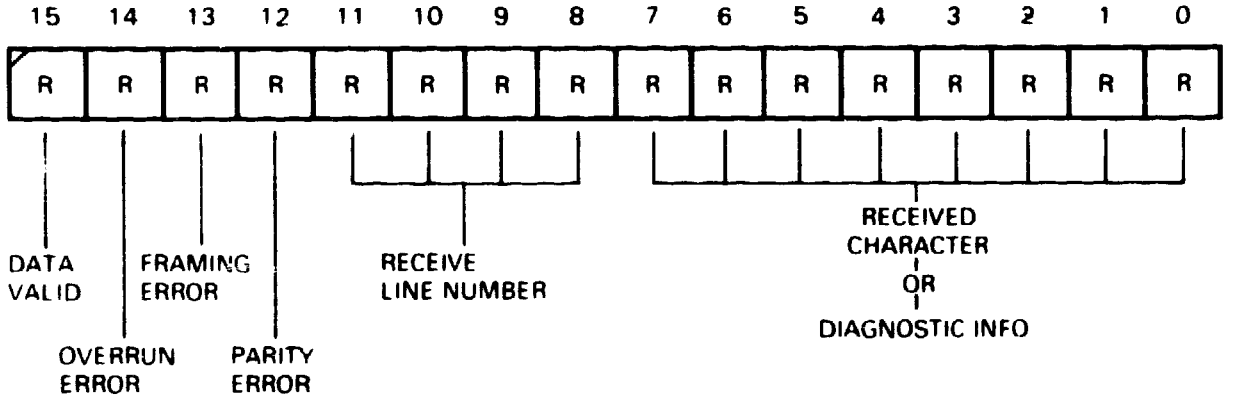
Bit	Name	Description
13	DIAG.FAIL (Diagnostic Fail) (R)	When set, this bit indicates that CXA16/CXB16 internal diagnostics have detected an error, either by the self-test diagnostic or by the BMP. This bit is associated with the diagnostic-passed LED. When it is set, the LED will be off. When it is cleared, the LED will be on. The bit is set by MASTER.RESET. It is cleared after the internal diagnostic programs have been run successfully. It is valid only after MASTER.RESET bit CSR<5> has been cleared.
12	TX.DMA.ERROR (Transmit DMA Error) (R)	If this bit is set and TX.ACTION is also set, either the channel indicated by CSR<11:8> has failed to transfer DMA data within 10 microseconds (in DHV11 mode), or 20 microseconds (in DHU11 mode), of bus request being acknowledged, or a memory parity error exists. The TBUFFAD1 and TBUFFAD2 registers will contain the address of the memory location at which the error occurred. TBUFFCT will be cleared.
<11:8>	TX.LINE (Transmit Line Number) (R)	If TX.ACTION is set, these bits hold the TX.ACTION line number.
7	RX.DATA.AVAIL (Received Data Available) (R)	When set, this bit indicates that a received character is available. The bit is clear when the receive FIFO is empty. It is used to request a receive interrupt. The bit is set after MASTER.RESET because the receive FIFO contains diagnostic information.

Bit	Name	Description
6	RXIE (Receiver Interrupt Enable) (R/W)	When set, this bit allows the CXA16/CXB16 to interrupt the host when RX.DATA.AVAIL is set. An interrupt is generated under the following conditions. 1. RXIE is set and a character is placed into the empty receive FIFO. 2. The receive FIFO contains one or more characters, and RXIE is changed from 0 to 1. It is cleared by BINIT but not by MASTER.RESET.
5	MASTER.RESET (Master Reset) (R/W)	This is set by the host in order to reset when in DHV11 mode. It stays set while the CXA16/CXB16 runs a self-test sequencer, and then performs an initialization sequence. The bit is then cleared to tell the host that the process is complete. This bit is set by BINIT (bus initialization signal), or by the host processor setting CSR<5>. If the CSR is written to while a MASTER.RESET sequence is in progress, the MASTER.RESET bit must be cleared.

Bit	Name	Description
4	SKIP (Skip Self-Test) (RW)	In DHU11 mode, this bit is used to shorten the reset/initialization time to about 30 milliseconds. The host program must only set this bit at the same time as it sets MASTER.RESET. It must then clear the bit, but must wait at least 20 microseconds before doing so. It is recommended that the host **always** set SKIP when setting MASTER.RESET. The CXA16/CXB16 will execute the full self-test, regardless of whether SKIP is set or not. The 1.7 seconds delay during MASTER.RESET is purely for DHU11 hardware compatibility. In DHV11 mode, this bit is ignored for compatibility reasons.
<3:0>	IND.ADDR.REG (Indirect Address Register) (R/W)	For indexed registers, these bits select one of 16 channels.

3.2.2.2 **Receive Buffer (RBUF)** -- A READ from base + 2 is interpreted by the CXA16/CXB16 hardware as a READ from the receive FIFO. Therefore, RBUF is a 256-character register with a single-word address. The least-significant bit (LSB) of the character is in bit 0.

RBUF (READ BASE + 2)



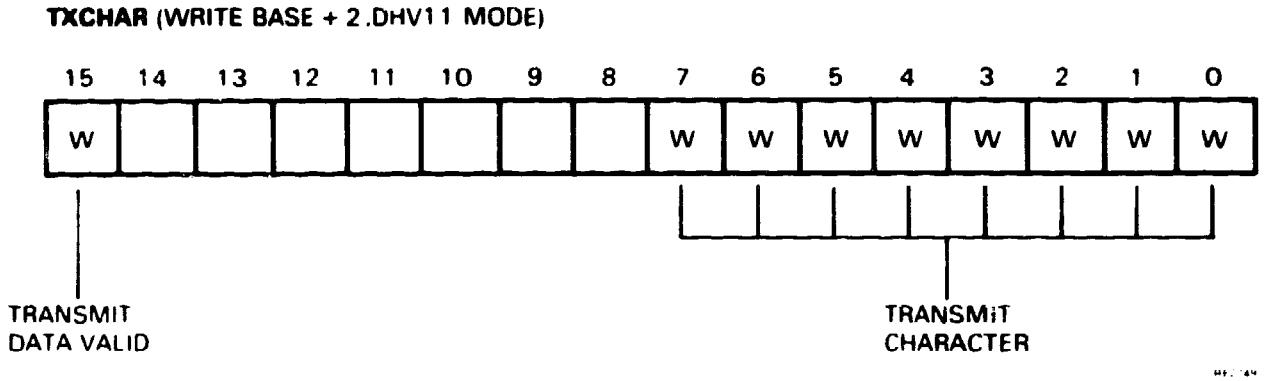
Bit	Name	Description
15	DATA.VALID (Data Valid) (R)	This bit is set if there is data in the receive FIFO. When this bit is clear, the contents of RBUF<14:0> are not valid. After self-test, diagnostic information is loaded into the receive FIFO. Therefore, this bit is always set after a successful master reset sequence.

Bit	Name	Description
14	OVERRUN.ERR (Overrun Error) (R)	This bit is set if one or more previous characters of the channel indicated by bits <11:8> were lost because of a full receive FIFO, or failure to service the UARTs (see also RX.CHAR). NOTE The 'all 1s' code for bits <14:12> is reserved. This code indicates that RBUF<7:0> holds diagnostic information.
13	FRAME.ERR (Framing Error) (R)	This bit is set if the first stop bit of the received character was not detected (also see RX.CHAR).
12	PARITY.ERR (Parity Error) (R)	This bit is set if this character has a parity error, and parity is enabled for the channel indicated by bits <11:8> (also see RX.CHAR).
<11:8>	RX.LINE (Receive Line Number) (R)	These bits hold the binary number of the channel on which the character of RBUF<7:0> was received.

Bit	Name	Description
<7:0>	RX.CHAR (Received Character) (R)	If RBUF<14:12> = 000, these eight bits contain the oldest character in the receive FIFO. The character is good. If RBUF<14:12> = 001, 010, or 011, these eight bits contain the oldest character in the receive FIFO, but the character is bad. If RBUF<14:12> = 111, then RBUF<7:1> contains diagnostic information. If there is an overrun condition, the UART data buffer for that channel will be cleared. This data will be lost. A null character is placed in the receive FIFO, and RBUF<14> is set. The CXA16/CXB16 does not have a break-detect bit. A line break is indicated to the program as a null character with FRAME.ERR set, and OVERRUN.ERR clear.

3.2.2.3 Transmit Character Register (TXCHAR) (DHV11 Mode Only) --

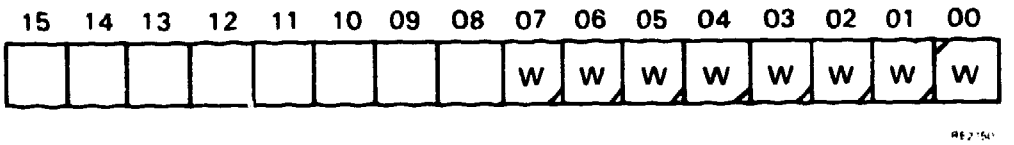
Single-character programmed transfers are made through the transmit character register. Bit function is as follows:



Bit	Name	Description
15	TX.DATA.VALID (Transmit Data Valid) (W)	When set, this bit instructs the CXA16/CXB16 to transmit the character held in bits <7:0>. The bit is sensed by the CXA16/CXB16, which then transfers the character, clears the bit, and sets TX.ACTION. TX.DATA.VALID and the character can be written together, or by separate MOVB instructions.
<7:0>	TX.CHAR (Transmit Character) (W)	This byte sets the character to be transmitted. The LSB is bit 0. For 7-, 6-, or 5-bit characters, unused bits must be 0.

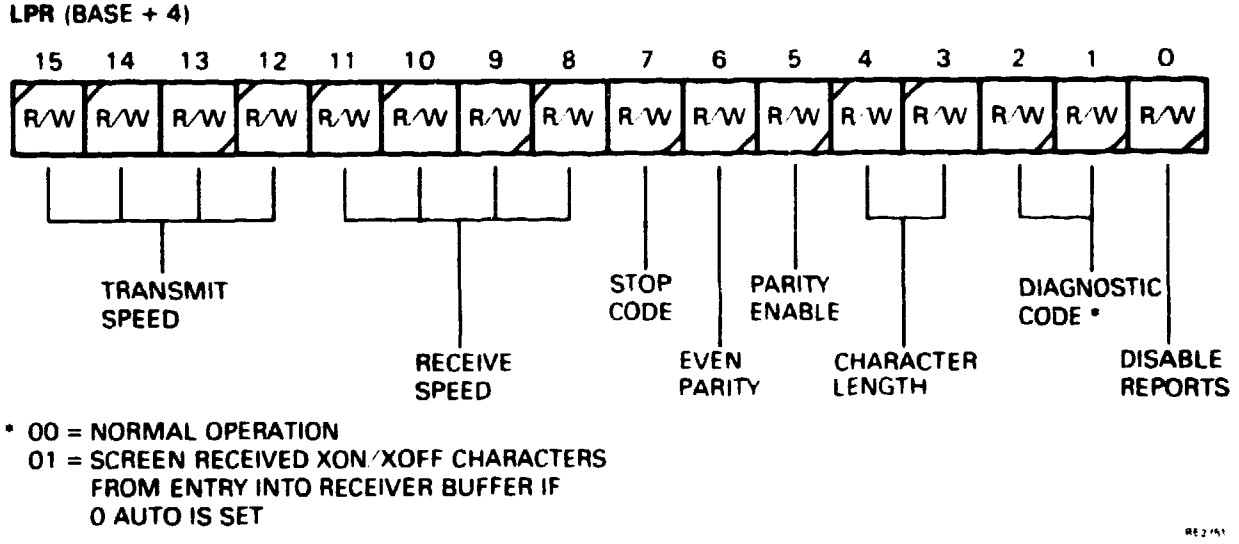
3.2.2.4 Receive Timer Register (RXTIMER) (DHU11 Mode Only) -- The indirect address register (CSR<3:0>) must = 0000 in order to access the receive timer. It can be used by the host to delay the receive interrupt.

Rx TIMER (WRITE BASE+2.DHU11 MODE)



Bit	Name	Description
<7:0>	RX.TIMER (Receive timer) (W BYTE)	The receive interrupt is normally raised when a received character is loaded into the previously empty receive FIFO. The binary number loaded into RXTIMER modifies this procedure as follows. 0 = Infinite timeout. This timeout will be overridden by the conditions below. 1 = No timeout. The interrupt will be raised immediately. 2 to 255 = Timer delay in milliseconds. The timer is overridden when the receive FIFO becomes three-quarters full (critical). Set to value of 1 by MASTER.RESET.

3.2.2.5 Line-Parameter Register (LPR) -- This register is used to configure its associated channel. Bit function is as follows:



Bit	Name	Description
<15:12>	TX.SPEED (Transmitted Data Rate) (R/W)	This is set to 1101 by MASTER.RESET (9600 bits/s). It defines the transmit data rate (Table 3-3).
<11:8>	RX.SPEED (Receive Data Rate) (R/W)	This is set to 1101 by MASTER.RESET (9600 bits/s). It defines the receive data rate (Table 3-3).
7	STOP.CODE (Stop Code) (R/W)	This bit defines the length of the transmitted stop bit. 0 = 1 stop bit for 5-, 6-, 7-, or 8-bit characters 1 = 2 stop bits for 6-, 7-, or 8-bit characters, or 1.5 stop bits for 5-bit characters The bit is cleared by MASTER.RESET.

Bit	Name	Description
6	EVEN.PARITY (Even Parity) (R/W)	If LPR<5> is set, this bit defines the type of parity. 1 = Even parity 0 = Odd parity The bit is cleared by MASTER.RESET.
5	PARITY.ENAB (Parity Enable) (R/W)	This causes a parity bit to be generated on transmit, and checked on receive. 1 = Parity enabled 0 = Parity disabled The bit is cleared by MASTER.RESET.
<4:3>	CHAR.LGTH (Character Length) (R/W)	This defines the length of characters. It does not include start, stop, and parity bits. 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits The is set to 11 by MASTER.RESET.

Bit	Name	Description
<2:1>	DIAG (Diagnostic Code) (R/W)	These are diagnostic control codes. They are used by the host as follows. 00 = Normal operation 01 = Causes the Background Monitor Program (BMP) to report the CXA16/CXB16 status to the receive FIFO.
<0>	DISAB.XRPT (Disable XON/XOFF Reporting) (R/W)	0 = XON and XOFF characters are reported on all channels. 1 = If LNCTRL<4> is also set for a particular channel, these characters are filtered from the received data stream, to relieve the host of the need to do so. On initialization, this bit is cleared. In order to read or write to this bit, CSR<3:0> must equal zero.

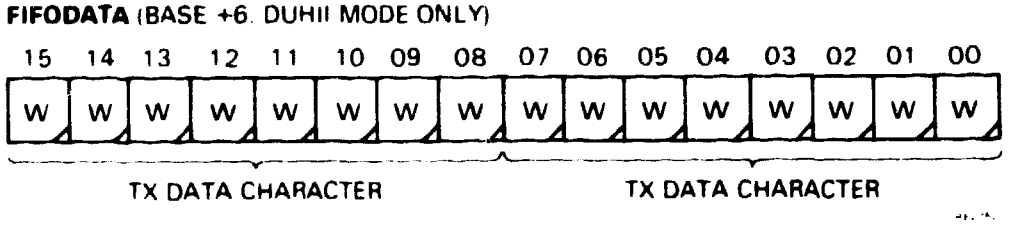
NOTE

**An XON code = 021(octal) = DC1 = CTRL/Q.
An XOFF code = 023(octal) = DC3 = CTRL/S.
No other codes are specified for the interface.**

Table 3-3 Data Rates

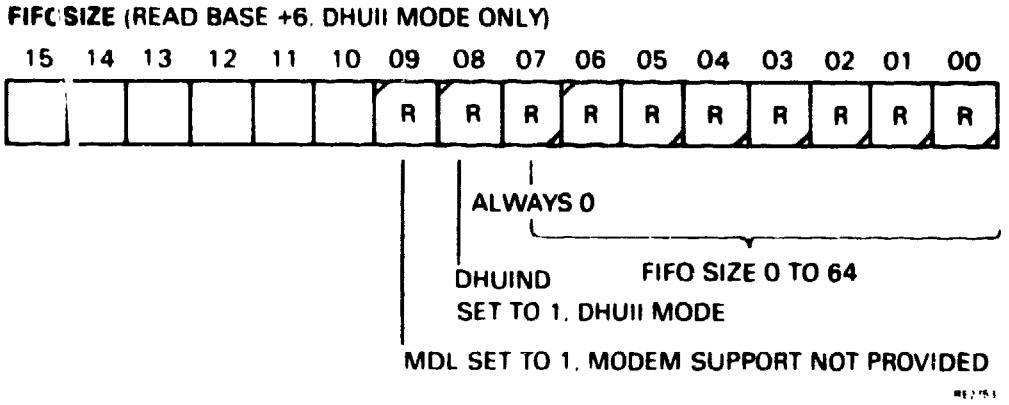
Code	Data Rate (Bits/s)	Maximum Error (%)
0000	50	0.01
0001	75	0.01
0010	110	0.08
0011	134.5	0.07
0100	150	0.01
0101	300	0.01
0110	600	0.01
0111	1200	0.01
1000	1800	0.01
1001	2000	0.19
1010	2400	0.01
1011	4800	0.01
1100	7200	0.01
1101	9600	0.01
1110	19200	0.01
1111	38400	0.01

3.2.2.6 FIFO Data Register (FIFODATA) (DHU11 Mode Only) -- To write a character or characters to a transmit FIFO, the host writes the character(s) to the FIFO data register of the appropriate channel. To make sure that there is room in the transmit FIFO, the host should first read the associated FIFO size register. If single characters are sent, they must be written to the low byte of FIFODATA.



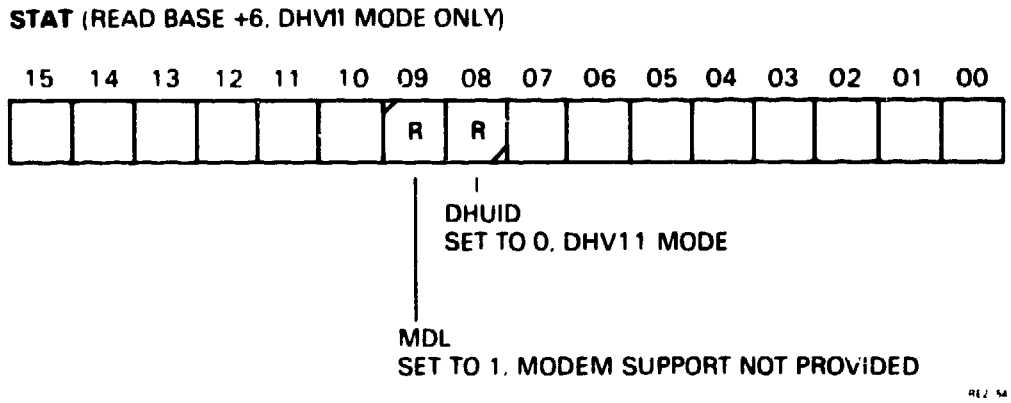
Bit	Name	Description
<15:0>	FIFODATA<15>	This word contains two characters (FIFO Data for transfer to the transmit FIFO). After a write-word action to this (W) register, FIFODATA<7:0> and then FIFODATA<15:8> are transferred to the FIFO. The least-significant bits of the characters are in FIFODATA bits 0 and 8. Unused bits must be cleared. This byte is cleared by MASTER.RESET.
<7:0>	FIFODATA<7> (FIFO Data Register) (W BYTE)	This byte contains a single character for transfer through to the transmit FIFO. After a write-byte action to this register, FIFODATA <7:0> is transferred to the FIFO. The least-significant bit of the character is in FIFODATA bit 0. Unused bits must be cleared. The byte is cleared by MASTER.RESET.

3.2.2.7 FIFO Size Register (FIFOSIZE) (DHU11 Mode Only) -- This low-byte register holds a number which indicates the space available in the transmit FIFO.



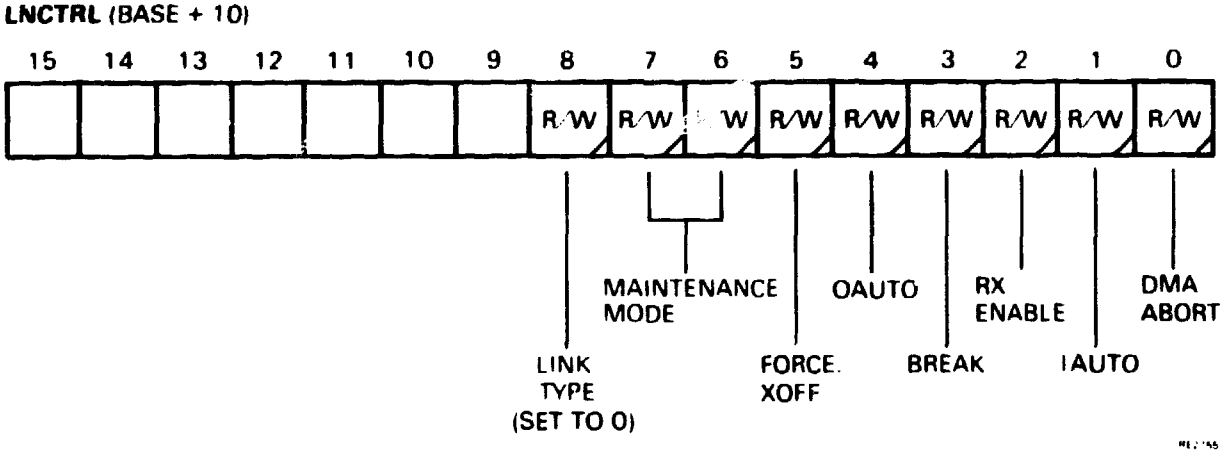
Bit	Name	Description
<9>	(MDL Modem Support Low) (R)	Always set to 1 - Modem support is not provided.
<8>	(DHUID Identification Bit) (R)	This bit allows software to distinguish between DHV11 mode and DHU11 mode. It is always 1 in DHU11 mode
<7:0>	FIFOSIZE (FIFO Size) (R BYTE)	This byte indicates the available space (in characters) in the transmit FIFO. The range is 00000000(binary) to 01000000(binary) (0(decimal) to 64(decimal)). This register should be read before sending a character, or a sequence of characters, to the FIFO data register. The byte is set to 64 characters by MASTER.RESET.

3.2.2.8 Line-Status Register (STAT) (DHV11 mode only) --



Bit	Name	Description
<9>	(MDL Modem Support Low) (R)	Always set to 1 - Modem support is not provided.
<8>	(DHUID Identification Bit) (R)	This bit allows software to distinguish between DHV11 mode and DHU11 mode. It is always 0 in DHV11 mode.

3.2.2.9 **Line-Control Register (LNCTRL)** -- The main function of this register is to control the line interface.



Bit	Name	Description
<7:6>	MAINT (Maintenance Mode) (R/W)	These bits can be written by the driver or test programs, in order to test the channel. The coding is as follows: 00 = Normal operation 01 = Automatic echo mode -- Received data is looped back to the terminal (regardless of the state of TX.ENA) at the data rate selected for the receiver. The received characters are processed normally and placed in the receive FIFO. Any data that the host attempts to transmit on this channel will be discarded by the OCTART. The RX.ENA bit must be set when operating in this mode.

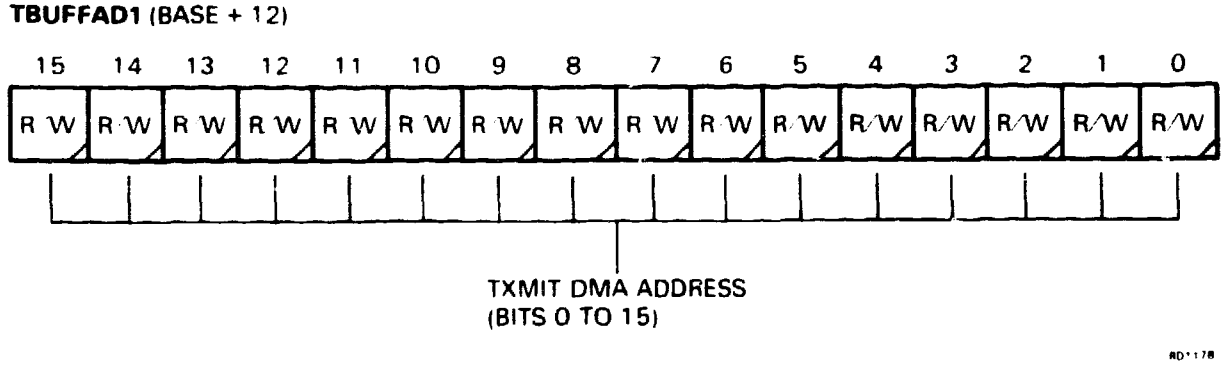
Bit	Name	Description
<7:6>	MAINT (cont.)	10 = Local loopback -- Data transmitted by the host is looped back to the receive buffer. Data received from the terminal is ignored, and the transmit data line to the terminal is held in the mark condition. The data rate selected for the transmitter is used for both transmission and reception. The TX.ENA bit still controls transmission in this mode. The RX.ENA bit is ignored. 11 = Remote loopback -- In this mode, data received from the terminal is looped back to the terminal at a clock rate equal to the received clock rate. The data is not placed in the receive FIFO. The state of TX.ENA is ignored. The RX.ENA bit must be set on this channel.
5	FORCE.XOFF (Force XOFF) (R/W)	This bit can be set by the program to indicate that this channel is congested at the host system (for example, if the typeahead buffer is full). When it sees this bit set, the CXA16/CXB16 will send an XOFF code. Until the bit is reset, XOFFs will be sent after every alternate character received on this channel. When the bit is reset, an XON will be sent unless IAUTO is set and the receive FIFO is critical.

Bit	Name	Description
4	OAUTO (Outgoing Auto Flow) (R/W)	This bit is the auto-flow control bit for outgoing characters. When set, if RX.ENA is also set, the CXA16/CXB16 will automatically respond to XON and XOFF codes received from a channel. The CXA16/CXB16 uses the TX.ENA bit in TBUFFAD2 to stop and start the flow. If DISAB.XRPT is also set, XON/XOFF codes are not entered in the receive FIFO.
3	BREAK (Break Control) (R/W)	If set, this bit forces the transmitter of this channel to the spacing state. Transmission is restarted when the bit is cleared. **NOTE** **If the line is idle, there may be a delay of up to 200 microseconds between writing the bit and the channel changing state. If a character is already being transmitted by the OCTART, the BREAK signal will be transmitted immediately afterwards.**
2	RX.ENA (Receiver Enable) (R/W)	If set, this receiver channel is enabled. If reset when this OCTART channel is assembling a character, that character is lost. The bit is cleared by MASTER.RESET.

Bit	Name	Description
1	IAUTO (Incoming Auto Flow) (R/W)	This is the auto-flow control bit for incoming characters. If it is set, the CXA16/CXB16 will control incoming characters by transmitting XON and XOFF codes. If the receive FIFO becomes more than three-quarters full, the CXA16/CXB16 will send an XOFF code to channels with this bit set. If a character is received, an XON will be sent when the receive FIFO becomes less than half full.
0	TX.ABORT (Transmit Abort) (R/W)	This bit is set by the driver program to halt data transmission. If a DMA transfer was in progress, the DMA address and count registers (TBUFFAD1, TBUFFAD2, and TBUFFCT) will be updated to reflect the number of characters which have been transmitted. The transfer can be continued by clearing TX.ABORT, and then setting TX.DMA.START in TBUFFAD2. No characters will be lost. If DMA is not in progress, the following actions will occur: DHV11 mode -- no action DHU11 mode -- characters in the transmit FIFO will be discarded. Because of sequencer delays, it is possible to transmit a few characters before the abort is actioned. Therefore, the host cannot determine how many characters have been lost.

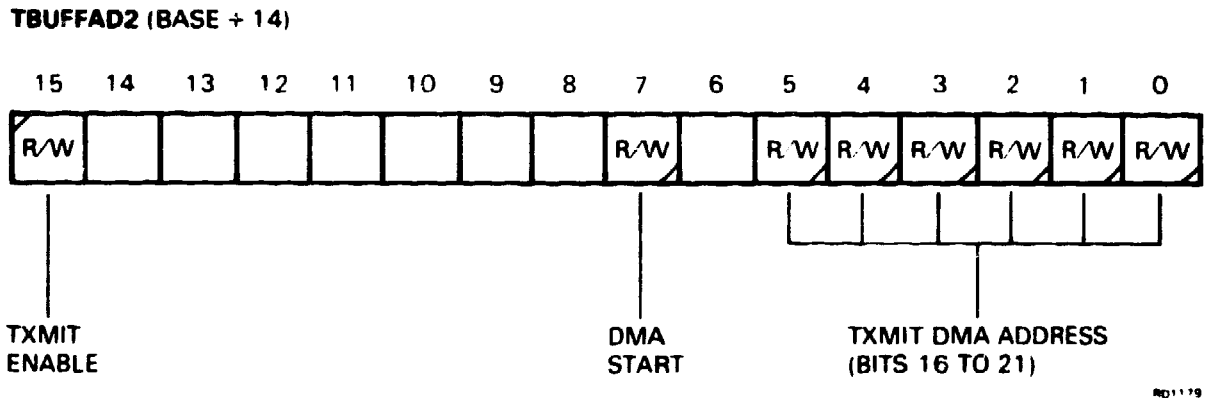
Bit	Name	Description
0	TX.ABORT (cont.)	When an abort sequence has been completed, the CXA16/CXB16 will set the TX.ACTION bit in the CSR. If the transmitter interrupt is enabled, the program will be interrupted at the transmit vector. The program must make sure that TX.ABORT is clear before setting TX.DMA.START. Otherwise, the transfer will be aborted before any characters are transmitted. The bit is cleared by MASTER.RESET.

3.2.2.10 Transmit Buffer Address Register Number 1 (TBUFFAD1) --



Bit	Name	Description
<15:0>	TBUFFAD<15:0> (Transmit Buffer Address [Low]) (R/W)	Bits <15:0> of the DMA address.

3.2.2.11 Transmit Buffer Address Register Number 2 (TBUFFAD2) --



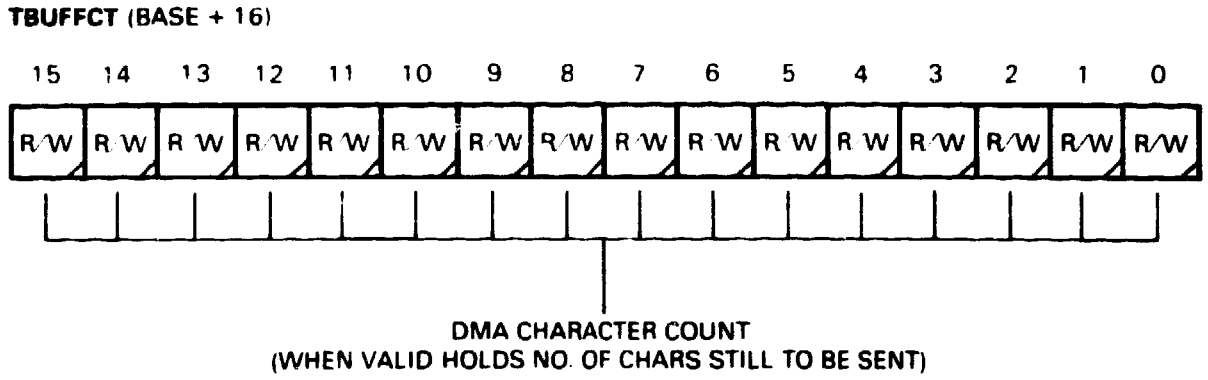
Bit	Name	Description
15	TX.ENA (Transmitter Enable) (R/W)	When this bit is set, the CXA16/CXB16 will transmit all characters. When cleared, the CXA16/CXB16 will only transmit internally generated flow-control characters. The bit is set by MASTER.RESET. In the OAUTO mode, this bit is used by the CXA16/CXB16 to control outgoing characters.
7	TX.DMA.START (Transmit DMA Start) (R/W)	This bit is set by the host to start a DMA transfer. The CXA16/CXB16 will reset the bit before returning TX.ACTION. The bit is cleared by MASTER.RESET.

NOTE

After setting this bit, the host must not write to TBUFFCT, TBUFFAD1, or TBUFFAD2 <7:0> until the TX.ACTION report has been returned.

Bit	Name	Description
<5:0>	TBUFFAD<21:16> (Transmit Buffer Address [High]) (R/W)	Bits <21:16> of the DMA address. Before a DMA transfer, TBUFFAD1 and the low byte of TBUFFAD2 are loaded with the start address of the DMA buffer. This address will be continuously changing during a DMA transfer and has no meaning. Once TX.ACTION has been returned, the register contains the final DMA transfer address.

3.2.2.12 Transmit DMA Buffer Counter (TBUFFCT) --



Bit	Name	Description
<15:0>	TX.CHAR.CT (Transmit Character Count) (R/W)	This word is loaded with the number of characters to be transferred by the DMA. The number of characters is specified as a 16-bit unsigned integer. After a DMA transfer has been aborted, this location will hold the number of characters still to be transferred. See also the previous NOTE.

3.3 PROGRAMMING FEATURES

3.3.1 Initialization

The CXA16/CXB16 is initialized by its on-board sequencers.

Initialization takes place after a bus reset sequence, or when the host sets CSR<5> (MASTER.RESET).

Before starting initialization, the on-board diagnostics run a self-test program. The results of this test are reported by eight diagnostic bytes in the receive FIFO.

The CXA16/CXB16 state, after a successful self-test, is as follows:

1. Eight diagnostic codes are placed in the receive FIFO
2. The diagnostic fail bit (CSR<13>) is reset
3. All channels set for:
    a. Send and receive 9600 bits/s
    b. Eight data bits
    c. One stop bit
    d. No parity
    e. Parity odd
    f. Auto-flow off
    g. Receive disabled
    h. Transmit enabled
    i. No break on line
    j. No loopback
    k. DMA character counter zero
    l. DMA start address register zero
    m. TX.DMA.START cleared
    n. TX.ABORT cleared
    o. AUTOFLOW REPORTS enabled

The CXA16/CXB16 clears the MASTER.RESET bit (CSR<5>) when initialization and self-test are complete.

3.3.2 Configuration

After CXA16/CXB16 self-initialization, the driver program can configure the CXA16/CXB16 as needed. This is done through the LPR and LNCTRL registers.

The line characteristics for a channel can be set up by writing to the LPR and LNCTRL registers associated with this channel. These are:

- Transmit speed
- Receive speed
- Number of stop bits
- Parity type or parity disabled
- Character length
- Flow control characteristics
- Normal or maintenance mode
- Receiver enable/disable

NOTE

If RX.ENA is reset while a received character is being assembled, that character will be lost.

3.3.3 Transmitting

Each CXA16/CXB16 channel can be set up to transfer the characters by DMA for transmission, or characters can be transferred to the module under program control.

3.3.3.1 **DMA Transfers** -- Before setting up the transfer of a DMA buffer, the program should make sure that TX.DMA.START is not set. TBUFFCT, TBUFFAD1, and TBUFFAD2 should not be written unless TX.DMA.START is clear.

Transmission starts when the program sets TX.DMA.START. DMA buffer size and start address can be written to TBUFFCT, TBUFFAD1, and TBUFFAD2 in any order. However, TBUFFAD2 contains TX.ENA and TX.DMA.START, so it is probably simpler to write to TBUFFAD2 last. By using byte operations on this register, TX.ENA and TX.DMA.START can be separated.

The CXA16/CXB16 will perform the transfer and set TX.ACTION when it is complete. If TXIE is set, the program will be interrupted at the transmit vector. Otherwise, TX.ACTION must be polled. TX.ACTION is not returned until the UART has completely transmitted the last character of the DMA buffer.

To abort a DMA transfer, the program must set TX.ABORT. The CXA16/CXB16 stops transmission and updates TBUFFCT, TBUFFAD1, and TBUFFAD2 <7:0> to reflect the number of characters transmitted. TX.DMA.START will be cleared. If TXIE is set, TX.ACTION interrupts the program at the transmit vector. If the program clears TX.ABORT and sets TX.DMA.START, transfer can be continued without loss of characters.

If a DMA transfer fails because of a memory error, the transmission will be terminated. TBUFFAD1 and TBUFFAD2 will point to the failing location. TBUFFCT will be cleared.

3.3.3.2 **Programmed I/O (DHV11 Mode)** -- Single characters are transferred through a channel TX.CHAR register. The character and the DATA.VALID bit must be written as defined in Section 3.2.2.3. Note that the character and the DATA.VALID bit can be written by separate MOVB instructions.

The CXA16/CXB16 returns TX.ACTION when the module removes the character from TX.CHAR. This will generate an interrupt if TXIE is set.

NOTE

In single-character mode, TX.ACTION is returned when the CXA16/CXB16 accepts the character, not when it has been transmitted. Each channel can buffer up to three characters. Therefore, if line parameters are changed immediately after the last TX.ACTION of a message, the end of the message could be lost unless three null characters are added to the end of each single-character programmed transfer message.

3.3.3.3 **Programmed I/O (DHU11 Mode)** -- Before writing a character or sequence of characters to the FIFODATA register, the program should read the FIFOSIZE register to check that there is space in the transmit FIFO.

If there is enough space, characters can be written as bytes (one character) or words (two characters) to FIFODATA. After a low-byte write, FIFODATA <7:0> is transferred to the FIFO. After a word write, FIFODATA <7:0> is transferred to the FIFO, followed by FIFODATA <15:8>. High-byte writes to FIFODATA are not allowed.

The CXA16/CXB16 returns TX.ACTION when the transmit FIFO becomes empty. An interrupt will also be generated if TXIE is set. As distinct from DMA mode, in programmed I/O mode TX.ACTION is returned when the CXA16/CXB16 transfers the last character from the transmit FIFO to the OCTART, not when it has been transmitted. Thus, if line parameters are changed immediately after the last TX.ACTION of a message, the end of the message could be lost. The program can avoid this loss by adding two null characters to the end of each programmed transfer FIFO message.

3.3.4 Receiving

Received characters, tagged with the channel number, error information and DATA.VALID, are placed in the receive FIFO. RX.DATA.AVAIL is clear when the receive FIFO is empty. When a character is put into the empty receive FIFO, the CXA16/CXB16 sets RX.DATA.AVAIL. A receive interrupt is generated if RXIE is set. RX.DATA.AVAIL stays set while there is valid data in the receive FIFO. It is recommended that the receive character routine continues to read characters from the receive FIFO until DATA.VALID is clear.

NOTE

The interrupt is dynamic. It is raised as RX.DATA.AVAIL is set after RXIE, or as RXIE is set after RX.DATA.AVAIL. If the interrupt routine does not empty the receive FIFO, RXIE must be toggled to raise another interrupt. In DHU11 mode, the interrupt is generated after a delay (set by RX.TIMER).

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

3.3.5 Interrupt Control

The CXA16/CXB16 provides one of two vector addresses during a bus interrupt sequence. The receive vector address is the address set up on the vector address switches. The transmit vector address is the receive vector address + 4.

The receive interrupt is generated when :

- RXIE is set and a character is placed into an empty receive FIFO
- RXIE is changed from 0 to 1, and the receive FIFO contains one or more characters.

NOTE

In DHU11 mode an interrupt is generated either immediately, or after the delay set by RX.TIMER.

The transmit interrupt vector is generated when:

- TXIE is set and TX.ACTION becomes set
- TXIE is changed from 0 to 1 while TX.ACTION is set

NOTE

The TX.ACTION FIFO can hold up to 16 TX.ACTION reports, but an interrupt is only generated when the FIFO receives a TX.ACTION after being empty. Therefore, it is recommended that your program reads the CSR until the TX.ACTION bit becomes clear. Reports will be lost if more than 16 have accumulated.

At the two vectors, the host must provide the addresses of suitable routines to deal with the above conditions.

In DHU11 mode, an interrupt is generated either immediately data is put into an empty receive FIFO, or after a delay, set by RX.TIMER.

3.3.6 Auto XON and XOFF

XON and XOFF characters are commonly used to control data flow on communications channels. To use this facility, interfaces must have suitable decoding hardware or software.

A channel using flow control that receives an XOFF stops sending characters until it receives an XON. A channel becoming overrun by received data sends an XOFF. It sends an XON when it can safely receive more data, if flow control is enabled.

The CXA16/CXB16 automatically controls character flow when programmed accordingly (auto-flow). Four bits control this function:

- IAUTO -- LNCTRL<1>
- FORCE.XOFF -- LNCTRL<5>
- OAUTO -- LNCTRL<4>
- DISAB.XRPT -- LPR<0>

IAUTO and FORCE.XOFF both control incoming characters. IAUTO is an enable bit which allows the level of the receive FIFO to control the generation of XOFF and XON characters. The FORCE.XOFF bit is a direct command from the program to control the incomimg data stream.

1. The CXA16/CXB16 hardware recognizes when the receive FIFO is three-quarters full and half full. The logic uses these states for auto-flow control.

   Each channel has a separate IAUTO bit. If there are 191 or more characters in the receive FIFO, and a character is received on a channel with IAUTO set, an XOFF character is sent. If the channel does not respond to the XOFF, the CXA16/CXB16 will send another XOFF in response to every alternate character received. An XON will be sent when the receive FIFO contains less than 128 characters, unless the FORCE.XOFF bit for that channel is set. XONs are only sent to channels to which an XOFF has previously been sent.

   By inserting XON and XOFF characters into the data stream, the program can perform flow control directly. However, if the CXA16/CXB16 is in IAUTO mode, the results will be unpredictable.

   In IAUTO mode, if RX.ENA is set, XON and XOFF characters will be transmitted even if TX.ENA is cleared.

2. When FORCE.XOFF is set, the CXA16/CXB16 sends an XOFF and then acts as if IAUTO is set and the receive FIFO is critical (was three-quarters full, and is not yet less than half full). When FORCE.XOFF is reset, an XON will be sent unless the receive FIFO is critical and IAUTO is set.

3. If the program sets OAUTO, the CXA16/CXB16 will automatically respond to XON and XOFF characters from the channel. It does this by clearing or setting the TX.ENA bit.

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

   Received XON and XOFF characters will always be reported through the receive FIFO, unless the DISAB.XRPT bit is set. It is possible, during read-modify-write operations by the program, for the CXA16/CXB16 to change the TX.ENA bit between the read and the write actions. For this reason, if DMA transfers are started while OAUTO is set, it is advisable to write to the low byte of TBUFFAD2 only.

4. If DISAB.XRPT is clear, XON and XOFF characters will be processed as normal characters and entered into the receive FIFO. DISAB.XRPT allows the individual line OAUTO bits to control whether XON or XOFF characters received on that channel are discarded. When DISAB.XRPT is set and OAUTO is set, this filtering is enabled.

NOTES

1. The CXA16/CXB16 may change the state of TX.ENA for up to 50 microseconds after OAUTO is cleared by the program.

2. When checking for flow-control characters, the CXA16/CXB16 only checks characters which do not contain transmission errors. The parity bit is stripped, and the remaining bits are checked for XON (21(octal)) and XOFF (23(octal)) codes.

3. Auto flow-control does not absolutely guarantee overrun errors will not occur. These errors may still occur if the transmitting devices do not respond to the XOFF immediately.

3.3.7 Error Indication

Four bits inform the program of transmission and reception errors.

- TX.DMA.ERR -- CSR<12>.
- PARITY.ERR -- RBUF<12>.
- FRAME.ERR -- RBUF<13>.
- OVERRUN.ERR -- RBUF<14>.

RBUF<14:12> may also identify a diagnostic code.

3.3.8 Maintenance Programming

The host can set bits 7 and 6 of LNCTRL to allow each channel to be configured in normal, automatic echo, local loopback, and remote loopback modes. These modes allow an individual data channel to be looped back to the host, or to be looped back to the terminal to assist in isolating communication problems.

The host must provide suitable software to use these modes.

3.3.9 Diagnostic Codes

3.3.9.1 Self-Test Diagnostic Codes -- After bus reset or master reset, the CXA16/CXB16 executes a self-test and initialization sequence. During the sequence, eight diagnostic codes are put in the receive FIFO, and RX.DATA.AVAIL is set.

After an error-free test, DIAG.FAIL will be reset and the 'diagnostic passed' LED will be on. If an error is detected, DIAG.FAIL will be set and the LED will be off.

3.3.9.2 Interpretation of Self-Test Codes -- The high byte of diagnostic codes in RBUF can be interpreted as in Section 3.2.2.2, except that bits <11:8> are not the line number. They indicate the sequence of the diagnostic byte, that is to say, 0 = first byte, 1 = second byte, and so on. Table 3-4 shows the meaning of each of the error codes.

Table 3-4 CXA16/CXB16 Self-Test Error Codes

Code (Octal) bits<7:0>	Test
201	Self-test null code (used as a filler)
203	Self-test skipped
211	Low OCTART error
213	High OCTART error
225	RAM error
All other error codes should be treated as an undefined error	
If bit 7 = 0 and bit 0 = 1, then bits <5:2> contain circuit revision information	
Bit 6 always reads 1 for the CXA16/CXB16, and indicates that the circuit contains control and OCTART chips	
Bit 1 indicates to which chip the information refers; 0 = control, 1 = OCTART	

After self-test, the eight FIFO codes consist of six diagnostic codes and two circuit revision codes. If there are less than six errors to report, null codes (201(octal)) fill the unused places.

After an error-free test, six null codes and two circuit revision codes will be returned.

Self-test may be 'skipped' to shorten the initialization cycle (see Section 3.3.9.3).

The module is still tested, even if self-test is skipped. The reset delay is much shorter, but test coverage is not affected; therefore skipping self-test is advantageous.

After 'skip self-test' self-test, the eight FIFO codes consist of six diagnostic codes and two circuit revision codes. If there are less than six errors to report, 203(octal) codes fill the unused places.

After an error-free test, six 203(octal) codes and two circuit revision codes will be returned.

3.3.9.3 Skipping Self-Test -- In DHU11 mode only, the method is to set SKIP (CSR bit 4) and MASTER.RESET (CSR bit 5) simultaneously, that is, write 60(octal) to the CSR register. SKIP must not be cleared until at least 20 microseconds after it was set. SKIP must be cleared by the host so that the master reset sequence can complete.

In DHV11 mode (this also works in DHU11 mode, but the previous method is preferred) the following method is used:

1. The program resets the CXA16/CXB16.

2. The program waits 10 ms (+ or - 1 ms) after issuing reset, and then writes 052525(octal) throughout the control registers (not the CSR) within the next 4 ms.

3. The self-test sequencer waits until 16 ms after reset, and then checks for a 052525(octal) code in common RAM. If it finds the code, the DIAG.FAIL bit is cleared, and the reset delay is shortened to 30 ms. If the code is not found, the reset delay is between one and two seconds.

NOTE

The program must not write to the CSR, or to the control registers, during the period starting 15 ms after reset, and ending when the MASTER.RESET bit is cleared. Writing during this period could cause a diagnostic fail condition.

3.3.9.4 Background Monitor Program (BMP) -- The CXA16/CXB16 BMP logic performs background self-tests by checking for OCTART interrupts. One or two codes are returned to the receive FIFO:

305(octal) -- CXA16/CXB16 running
307(octal) -- CXA16/CXB16 defective (also LED off)

A single diagnostic word is returned to the receive FIFO. The low byte contains the diagnostic code. In the high byte, OVERRUN.ERR, FRAME.ERR, and PARITY.ERR are all set to indicate that bits <7:0> do not hold a normal character. The line number (RBUF<11:8>) = 0.

BMP normally only reports when it finds an error. However, the program can get a BMP report at any time to validate the CXA16/CXB16. This is done by setting DIAG (LPR <2:1>) of any channel to 01. The line number returned is that of the LPR used to request the report.

On completing the check, BMP clears the 01 code in DIAG. The host should not write to the LPR of that channel until DIAG is cleared to 00.

3.4 PROGRAMMING EXAMPLES

This section contains programming examples of the CXA16/CXB16 used in DHU11 mode. Programs for DHV11 mode are included. These programs are not presented as the only way of driving the option, and are neither guaranteed nor supported.

3.4.1 Resetting the CXA16/CXB16

In the following example:

- DIAG is a routine to check the diagnostic codes. It returns with CARRY set if it detects an error code.
- The loop at 1$ can take up to 2.5 seconds, so the programmer could poll by using a timer or poll at interrupt level zero.

```
;
; A ROUTINE TO RESET THE CXA16/CXB16 AND CHECK THAT IT IS
; FUNCTIONING CORRECTLY.
;

CXARES::
        MOV     #40,@#CXACSR            ; SET MASTER.RESET AND
                                        ; CLEAR INTERRUPT ENABLES.
1$:     BIT     #40,@#CXACSR            ; WAIT FOR MASTER.RESET TO
        BNE     1$                      ; CLEAR.
        BIT     #20000,@#CXACSR         ; CHECK THE DIAGNOSTICS FAIL
        BNE     DIAGER                  ; BIT.
                                        ; NOTE:TEST INSTRUCTION IS
                                        ; OK BECAUSE THERE ARE
                                        ; NO TX.ACTS PENDING.
        MOV     #8.,R5                  ; SET UP A COUNT.
                                        ;
2$:     MOV     @#RBUFF,R0              ; GET NEXT DIAGNOSTIC CODE.
        JSR     PC,DIAG                 ; PROCESS IT.
        BCS     DIAGER                  ; CARRY SET - MUST HAVE BEEN
                                        ; AN ERROR.
        SOB     R5,2$                   ; GO BACK FOR NEXT CODE.

        RTS     PC                      ; RETURN - CARD IS RESET.

;
; CXA16/CXB16 HAS FAILED TO RESET PROPERLY, SO HALT AND WAIT FOR
; THE FIELD SERVICE ENGINEER.
;

DIAGER: HALT
        BR      DIAGER
```

3.4.2 Configuration

This routine sets the characteristics of channel 1 as follows:

1. Transmit and receive at 300 bits/s
2. Seven data bits with even parity and one stop bit
3. Transmitters and receivers enabled
4. No automatic flow control.

```
SETUP::
        MOV     #1,@#CXACSR             ; LOAD INDEX REG
                                        ; WITH CHANNEL NO.
        MOV     #052560,@#LPR           ; DATA RATE, STOP BITS,
                                        ; PARITY AND LENGTH.
        MOV     #4,@#LNCTRL             ; ENABLE THE RECEIVER.
        MOVB    #200,@#TBFAD2+1         ; ENABLE THE TRANSMITTER.

        RTS     PC                      ; RETURN - CHANNEL 1 DONE.
```

3.4.3 Transmitting

3.4.3.1 Programmed Transfer (DHU11 Mode) -- The following is a program to send a message on channel 1.

The CSR is polled for TX.ACTION reports, but a TX.ACTION interrupt could also be used.

This program would function on a CXA16/CXB16 with only this channel active. Otherwise it would lose TX.ACTION reports of other channels. However, a program to control all channels would be too big to use as an example.

```
;
; A ROUTINE TO WRITE A MESSAGE TO CHANNEL 1 USING FIFO OUTPUT
; MODE (PROGRAMMED TRANSFERS).
;

FIFOUT::
        MOV     #1,@#CXACSR         ; POINT TO CHANNEL WE WISH
                                    ; TO TALK TO.
        MOV     #MESS,R0            ; POINT TO MESSAGE.
        MOV     #MESIZE,R1          ; PUT COUNT IN.
1$:
        TSTB    @#FIFOSIZE          ; CHECK THAT THERE IS SPACE IN
        BEQ     1$                  ; THE FIFO.
        MOVB    (R0)+,@#FIFODATA    ; MOVE CHARACTER TO TRANSMIT FIFO.
        SOB     R1,1$               ; GO BACK FOR NEXT CHARACTER.
2$:
        MOV     @#CXACSR,R2         ; WAIT FOR TX.ACT.
        BPL     2$

        BIC     #170377,R2          ; ISOLATE CHANNEL NUMBER.
        CMP     #000400,R2
        BNE     2$                  ; IGNORE THE TX.ACT IF IT IS
                                    ; NOT OURS (SHOULD NOT HAPPEN).

        RTS     PC                  ; MESSAGE SENT.

MESS:   .ASCII  /A TRANSMIT FIFO MESSAGE FOR CHANNEL 1/
MESIZE  =       .-MESS
        .EVEN
```

3.4.3.2 Single-Character Programmed Transfer (DHV11 Mode) -- This is a program to send a message on channel 1. The message (MESS) is an ASCII string with a null character as terminator.

Polling is used, but a TX.ACTION interrupt could also be used.

This program would function on a CXA16/CXB16 with only this channel active. Otherwise it would lose TX.ACTION reports of other channels. However, a program to control all channels would be too big to use as an example.

```
;
; A ROUTINE TO WRITE A MESSAGE TO CHANNEL 1 USING SINGLE-CHARACTER
; MODE.
;

SINGOT::
        MOV     #1,@#CXACSR             ; LOAD INDEX REG WITH
                                        ; CHANNEL NO.
        MOV     #MESS,R0                ; POINT TO MESSAGE.
1$:
        MOVB    (R0)+,@#TXCHAR          ; MOVE CHARACTER TO TRANSMIT
                                        ; BUFFER.
        BEQ     3$                      ; GO RETURN IF ALL CHARACTERS
                                        ; GONE.
        MOVB    #200,@#TXCHAR+1         ; SET DATA VALID BIT TO START.
2$:
        MOV     @#CXACSR,R1             ; WAIT FOR TX.ACT
        BPL     2$

        BIC     #174377,R1              ; ISOLATE CHANNEL NUMBER.
        CMP     #000400,R1
TXI     BNE     2$                      ; IGNORE THE TX.ACT IF IT IS
                                        ; NOT OURS (SHOULD NOT HAPPEN)
        BR      1$                      ; GO BACK FOR NEXT CHARACTER.

3$:
        RTS     PC                      ; MESSAGE SENT.

MESS:   .ASCIZ /A SINGLE-CHARACTER MESSAGE FOR CHANNEL 1/
```

3.4.3.3 DMA Transfer --

```
;
; THIS PROGRAM SENDS A MESSAGE OUT ON EACH LINE OF THE CXA16/CXB16
; AND HALTS THE MACHINE WHEN ALL TRANSMISSIONS HAVE COMPLETED.
;
; THE MESSAGES ARE TRANSMITTED USING DMA MODE, AND INTERRUPTS ARE
; USED TO SIGNAL TRANSMISSION COMPLETION.
;

DMAINT::
        MOV     #TXINT,@#TXVECT     ; SET UP THE INTERRUPT VECTORS.
        MOV     #200,@#TXPSW        ; INTERRUPT PRIORITY FOUR.
        MOV     #16.,R0             ; SIXTEEN LINES TO START.
        CLR     R1                  ; START AT LINE ZERO.
1$:
        MOVB    R1,@#CXACSR         ; SELECT THE REGISTER BANK.
        MOV     #DMASIZ,@#TBFCNT    ; SET LENGTH OF MESSAGE.
        MOV     #DMAMES,@#TBFAD1    ; SET LOWER 16 ADDRESS BITS.
        MOV     #100200,@#TBFAD2    ; START DMA WITH TRANSMITTER
                                    ; ENABLED (ASSUME UPPER ADDRESS
                                    ; BITS ARE ZERO).
        INC     R1                  ; POINT TO NEXT CHANNEL.
        SOB     R0,1$               ; REPEAT FOR ALL LINES.

        CLR     R5                  ; R5 IS USED BY INTERRUPT ROUTINE.
        MOVB    #100,@#CXACSR+1     ; ENABLE TRANSMITTER INTERRUPTS.
2$:
        CMP     #16.,R5             ; WAIT FOR ALL LINES TO FINISH.
        BNE     2$
3$:
        HALT                        ; ALL DONE, SO STOP.
        BR      3$
```

```
;
; TRANSMITTER INTERRUPT ROUTINE.
;
; R5 IS INCREMENTED AS EACH LINE COMPLETES.
;

TXINT::
        MOV     @#CXACSR,R0     ; GET LINE NUMBER OF FINISHED LINE.
        BIT     #100000,R0      ; CHECK FOR (ANOTHER) TX.ACTION.
        BEQ     4$              ; IF NOT, GO RETURN AND WAIT.

        BIT     #010000,R0      ; CHECK FOR DMA FAILURE.
        BNE     5$              ; GO HALT - MEMORY PROBLEM.
                                ; NOT USED IN DHV11 MODE.

        INC     R5              ; FLAG THAT ANOTHER LINE HAS FINISHED.
        BR      TXINT
4$:
        RTI
5$:
        HALT                    ; MEMORY PROBLEM.
        BR      5$

DMAMES: .ASCII  <15><12><7><7><7>/SYSTEM CLOSING DOWN NOW/
DMASIZ  =       .-DMAMES
        .EVEN
```

3.4.3.4 Aborting a Transmission --

```
;
; THIS ROUTINE IS CALLED TO ABORT A TRANSMISSION (EITHER DMA OR
; FIFO) IN PROGRESS ON A SPECIFIED LINE. THIS ROUTINE MAKES THE
; (RATHER RASH) ASSUMPTION THAT THERE ARE NO OTHER TRANSFERS IN
; PROGRESS.
;
; ON ENTRY, R0 CONTAINS THE NUMBER OF THE LINE TO BE ABORTED.
;

TXABRT::
        MOV     R0,@#CXACSR     ; POINT TO THE CHANNEL TO BE ABORTED.
        BIS     #1,@#LNCTRL     ; SET THE TRANSMIT ABORT BIT.
1$:
        MOV     @#CXACSR,R1     ; WAIT FOR THE TX.ACT.
        BPL     1$
        SWAB    R1              ; CHECK IT IS OUR LINE.
        BIC     #177760,R1
        CMP     R0,R1
        BNE     1$              ; IGNORE IT IF IT IS NOT (OUR
                                ; ASSUMPTION WAS WRONG!)

        BIC     #1,@#LNCTRL     ; CLEAR DOWN THE ABORT FLAG
                                ; FOR NEXT TIME.

        RTS     PC              ; BUFFER COMPLETELY ABORTED,
                                ; IF A DMA WAS IN PROGRESS, THE
                                ; DMA REGISTERS REFLECT WHERE
                                ; THE CXA16/CXB16 HAD GOT TO.
```

3.4.4 Receiving

```
;
; THIS ROUTINE PROCESSES RECEIVED CHARACTERS UNDER INTERRUPT CONTROL.
; IF AN XOFF IS RECEIVED, THE TRANSMITTER FOR THAT CHANNEL IS TURNED
; OFF. IF AN XON IS RECEIVED, THE TRANSMITTER IS TURNED BACK ON.  ALL
; OTHER CHARACTERS ARE IGNORED.
;

RXAUTO::
        MOV     #RXINT,@#RXVECT     ; SET UP THE INTERRUPT VECTORS.
        MOV     #200,@#RXPSW        ; INTERRUPT PRIORITY FOUR.
        MOV     #16.,R0             ; ENABLE ALL THE RECEIVERS,
        CLR     R1                  ; STARTING AT CHANNEL ZERO.
1$:
        MOVB    R1,@#CXACSR         ; SELECT THE LINE.
        BIS     #4,@#LNCTRL         ; ENABLE THIS RECEIVER.
        INC     R1                  ; SET POINTER TO NEXT CHANNEL.
        SOB     R0,1$

        MOVB    #0,@#CXACSR         ; SELECT CHANNEL ZERO.
        MOV     #20.,@#RXTIMR       ; SET DELAY TO 20MS.
                                    ; NOT USED IN DHV11 MODE.

        MOVB    #100,@#CXACSR       ; ENABLE THE RECEIVER INTERRUPTS.

        RTS     PC                  ; RETURN -
                                      INTERRUPTS DO THE RESET.
```

```
;
; INTERRUPT ROUTINE TO DO THE MAIN TASK.
;

RXINT::
        MOV     R0,-(SP)        ; SAVE CALLER'S REGISTERS.
RXNXTC:
        MOV     @#RBUFF,R0      ; GET THE CHARACTER.
        BPL     RXIEND          ; IF NO DATA VALID, WE HAVE FINISHED.
        MOV     R0,-(SP)        ; CHECK FOR ERRORS AND
        BIC     #107777,(SP)+   ; DIAGNOSTICS CODES.
        BNE     RXNXTC          ;  - JUST IGNORE THEM (BAD PRACTICE).

        BIC     #170200,R0      ; REMOVE UNNECESSARY BITS.
        SWAB    R0              ; POINT TO THIS CHARACTER'S LINE.
        BIS     #100,R0         ; (ADD THE INTERRUPT ENABLE BIT.)
        MOVB    R0,@#CXACSR
        SWAB    R0              ; PUT CHARACTER BACK IN LOWER BYTE.
        CMPB    #21,R0          ; WAS IT AN "XON"?
        BNE     1$              ; NO - GO CHECK FOR AN "XOFF"

        BISB    #200,@#TBFAD2+1 ; ENABLE THE TRANSMITTER.
        BR      RXNXTC          ; GO CHECK FOR MORE CHARACTERS.
1$:
        CMPB    #23,R0          ; WAS IT AN "XOFF"?
        BNE     RXNXTC          ; NO - GO CHECK FOR MORE CHARACTERS.

        BICB    #200,@#TBFAD2+1 ; DISABLE THE TRANSMITTER.
        BR      RXNXTC          ; GO CHECK FOR MORE CHARACTERS.

RXIEND:
        MOV     (SP)+,R0        ; RESTORE THE DESTROYED REGISTER.
        RTI
```

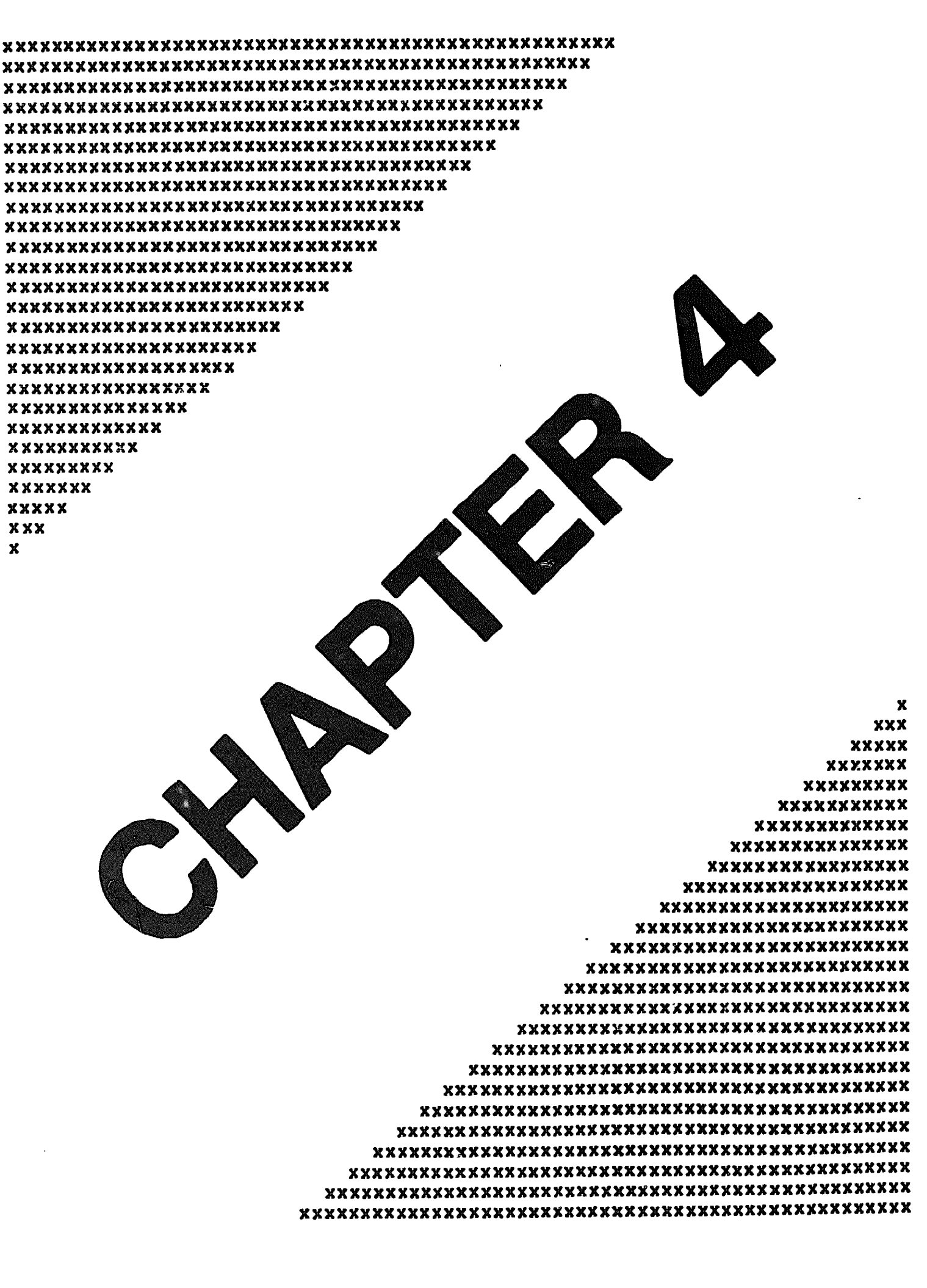
3.4.5 Auto XON and XOFF

```
;
; THIS PROGRAM SENDS A MESSAGE OUT ON EACH LINE OF THE CXA16/CXB16
; AND HALTS THE MACHINE WHEN ALL TRANSMISSIONS HAVE COMPLETED.
;
; THE MESSAGES ARE TRANSMITTED USING DMA MODE, AND INTERRUPTS ARE
; USED TO SIGNAL TRANSMISSION COMPLETION.
;
; AUTOMATIC FLOW CONTROL IS ENABLED ON THE OUTGOING DATA.
;

TXAUTO::
        MOV     #ATOINT,@#TXVECT    ; SET UP THE INTERRUPT VECTORS.
        MOV     #200,@#TXPSW        ; INTERRUPT PRIORITY FOUR.
        MOV     #16.,R0             ; SIXTEEN LINES TO START.
        CLR     R1                  ; START AT LINE ZERO.
1$:
        MOVB    R1,@#CXACSR         ; SELECT THE REGISTER BANK.
        BIS     #24,@#LNCTRL        ; ENABLE AUTOMATIC FLOW CONTROL
                                    ; ON THE TRANSMITTED DATA.
        MOV     #AUTOSZ,@#TBFCNT    ; SET LENGTH OF MESSAGE.
        MOV     #AUTOMS,@#TBFAD1    ; SET LOWER 16 ADDRESS BITS.
        MOV     #100200,@#TBFAD2    ; START DMA WITH TRANSMITTER
                                    ; ENABLED (ASSUME UPPER ADDRESS
                                    ; BITS ARE ZERO).
        INC     R1                  ; POINT TO NEXT CHANNEL.
        SOB     R0,1$               ; REPEAT FOR ALL LINES.

        CLR     R5                  ; R5 IS USED BY INTERRUPT ROUTINE.
        MOVB    #100,@#CXACSR+1     ; ENABLE TRANSMITTER INTERRUPTS.
2$:
        CMP     #16.,R5             ; WAIT FOR ALL LINES TO FINISH.
        BNE     2$
3$:
        HALT                        ; ALL DONE, SO STOP.
        BR      3$
```

```
;
; TRANSMITTER INTERRUPT ROUTINE.
;
; R5 IS INCREMENTED AS EACH LINE COMPLETES.
;

ATOINT::
        MOV     @#CXACSR,R0     ; GET LINE NUMBER OF FINISHED LINE.

        BPL     2$              ; GO RETURN IF NO MORE TX.ACTIONS.
                                ; NOT USED IN DHV11 MODE.

        BIT     #10000,R0       ; CHECK FOR DMA FAILURE.
        BNE     4$              ; GO HALT - MEMORY PROBLEM.

        INC     R5              ; FLAG THAT ANOTHER LINE HAS FINISHED.

        BR ATOINT               ; CHECK FOR MORE TX.ACTIONS.
                                ; NOT USED IN DHV11 MODE.

2$:
        RTI
4$:
        HALT                    ; MEMORY PROBLEM
        BR      4$

AUTOMS: .ASCII  <15><12><7><7><7>/SYSTEM CLOSING DOWN NOW/
AUTOSZ  =       .-AUTOMS
        .EVEN
```

3.4.6 Checking Diagnostic Codes

```
;
; THIS ROUTINE CHECKS THE DIAGNOSTICS CODES RETURNED FROM THE
;
; CXA16/CXB16 ON ENTRY, R0 CONTAINS THE CHARACTER RECEIVED FROM THE
;
; CXA16/CXB16. ON EXIT, THE CARRY BIT WILL BE CLEAR FOR SUCCESS,
;
; SET FOR FAILURE.

DIAG::
        MOV     R0,-(SP)        ; SAVE THE CODE FOR LATER.

        BIC     #107776,R0      ; CHECK THAT IT IS A DIAG. CODE.
        CMP     #070001,R0
        BNE     DIAGEX          ; IF NOT, JUST EXIT NORMALLY.

        MOV     (SP),R0         ; GET THE CODE BACK.

        BITB    #200,R0         ; CHECK FOR ROM VERSION NUMBER.
        BEQ     DIAGEX
        CMPB    #201,R0         ; SELF-TEST NULL CODE.
        BEQ     DIAGEX
        CMPB    #203,R0         ; SELF-TEST SKIPPED CODE.
        BEQ     DIAGEX

        CMPB    #305,R0         ; CXA RUNNING CODE.
        BEQ     DIAGEX
                                ; ALL THE REST ARE ERROR CODES.

        SEC                     ; AN ERROR CODE WAS RECEIVED, SO
        BR      DIAGXX          ; SET THE CARRY FLAG.

DIAGEX:
        CLC                     ; EVERYTHING OK, SO CLEAR CARRY.
DIAGXX:
        MOV     (SP)+,R0        ; RESTORE THE CHARACTER/INFO.
        RTS     PC
```


CHAPTER 4

CHAPTER 4
TROUBLESHOOTING

4.1 SCOPE

This chapter explains how to isolate the cause of a communications problem between the CXA16/CXB16 and the equipment to which it is connected.

4.2 TROUBLESHOOTING

4.2.1 Preventive Maintenance

No preventive maintenance is needed for this option. However, you should always ensure that all cables are clear of danger, and that all the connectors are secure.

Make sure that all cables are clearly labeled, so that you can easily identify which channel number and which CXA16/CXB16 module is associated with each terminal.

4.2.2 Troubleshooting Procedures

Troubleshooting procedures are to identify whether the problem is caused by:

- The module
- A terminal
- The cabling.

First, decide whether the problem is associated with one channel, a group of 8 channels, or all 16 channels.

If all channels are faulty, run the user diagnostics to test the module. Also check whether your software has a driver for the CXA16/CXB16.

If a group of eight channels is faulty, check the BC16D-25 cables between the module and the cable concentrator.

For single-channel problems:

1. Check for loose cables and connectors.

2. Verify that the terminal is working correctly. If necessary, swap it with another one.

3. Verify that the signals from the terminal are reaching the CXA16/CXB16 by using the H3101 test connector.

   a. Check which CXA16/CXB16 line is suspect and disconnect the appropriate 36-way connector at the CXA16/CXB16 bulkhead.

   NOTE

   This interrupts the other seven lines using that connection.

   b. Connect the cable to the H3101 loopback connector (see Figure 4-1).

   c. Type characters at the terminal connected to the suspect line. If characters are echoed back when the H3101 is connected, the cable and terminal are working.

4. Rectify the cable or terminal fault, if there is one. If not, make sure that the user diagnostics for the module run correctly, and that the communication parameters selected for that channel match the terminal's characteristics.

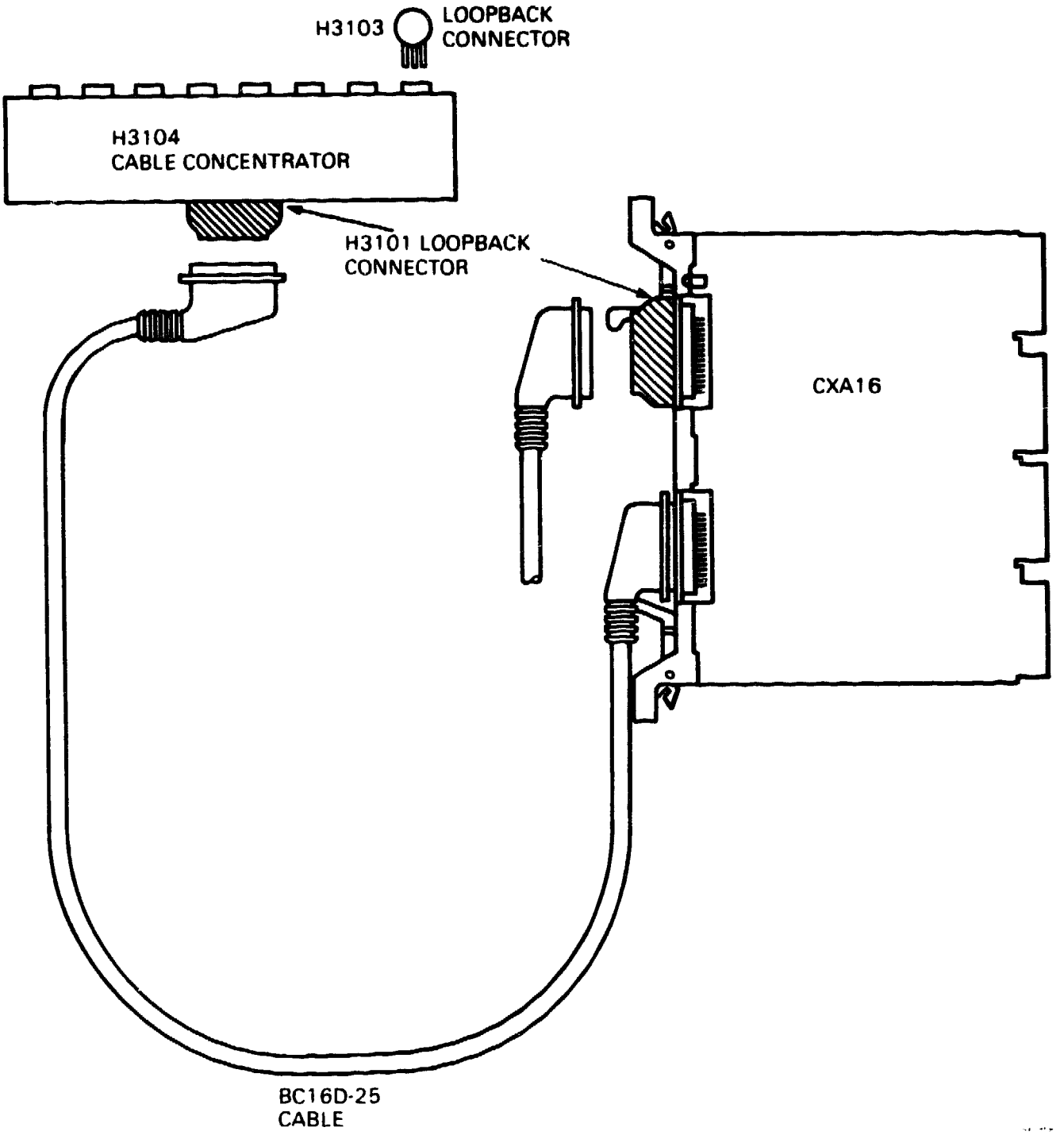


Figure 4-1 Cable Loopbacks

4.3 INTERNAL DIAGNOSTICS

Internal diagnostics run without intervention from the operator. There are two tests: the self-test and the background monitor program (BMP).

4.3.1 Self-Test

The self-test starts immediately after the Q-bus or module has been reset. It performs a comprehensive internal logic test, but does not test the Q-bus interface or serial-line drivers and receivers, The self-test gives a GO/NO-GO indication via the DIAG.FAIL bit and the 'diagnostics passed' LED on the module. The self-test also reports error or status information to the host via the receive FIFO.

The self-test has completed successfully if the LED is ON two seconds after the self-test has been initiated, either by powering up the module or by resetting the module through the program interface. The LED is turned off while the self-test sequencer is executing. The OFF period varies depending on whether DHV11 or DHU11 mode was selected, and on whether the self-test was invoked using the self-test skip feature of the program interface; but it will not exceed two seconds.

Self-test provides a high level of confidence that the majority of the module logic is working. The user diagnostics must also be used to test the Q-bus interface and verify that the switch settings on the module switchpacks are correct.

4.3.2 Background Monitor Program (BMP)

When the CXA16/CXB16 is not doing other tasks, the BMP carries out tests on the module. If an error is detected, the BMP reports to the host via the receive FIFO, and also switches OFF the 'diagnostics passed' LED.

By writing codes to the line-parameter register, the host can cause the BMP to report the status of the device, even if an error has not been detected. This facility is used if the host suspects that the option is unavailable.

NOTE

More information on the self-test and BMP diagnostics is given in Chapter 3 of this manual.

4.4 MicroPDP-11 DIAGNOSTICS

4.4.1 User-Mode Diagnostics

These tests can be used by an untrained operator to verify the basic operation of the option. User-mode tests do not cause any disruption to data networks or devices to which the CXA16/CXB16 may be connected. Such networks and devices do not have to be disconnected from the CXA16/CXB16 during the tests. The MicroPDP-11 system manuals describe how to load and run these diagnostics.

4.4.1.1 Running User-Mode Tests -- All user-mode tests are run by selection from the test menu displayed when the user diagnostics are booted. See Chapter 2 for more details.

4.5 MicroVAX II DIAGNOSTICS

Diagnostics for MicroVAX II systems all run under the MicroVAX Maintenance System (MMS). The MicroVAX II system manuals describe how to load the MMS into the MicroVAX II, and how to run MMS diagnostics. All the tests can be run by selection from the test menus displayed when MMS is booted.

4.5.1 User-Mode Tests

These tests can be used by an untrained operator to verify the basic operation of the option. User-mode tests do not cause any disruption to data networks or devices to which the CXA16/CXB16 may be connected. Such networks and devices do not have to be disconnected from the CXA16/CXB16 during the tests. See Chapter 2 for more details.

4.6 MicroPDP-11 SERVICE MODE DIAGNOSTICS

4.6.1 MicroPDP-11 Functional Diagnostic

The functional diagnostic for a MicroPDP-11 is VHQA**.BIN. This also tests other modules in the same family, for example, DHV11.

After booting, the program asks you four questions about the hardware configuration; these are:

1. CSR address
2. Vector address
3. Active line bitmap
4. Loopback type

The diagnostic will then 'size' the option to determine operating mode (DHU11/DHV11), modem or data-leads only, 8 or 16 lines, and then print this information. The control chip and OCTART revision levels are also printed. Always check that the unit 'sizes' correctly.

4.6.1.1 Test Summaries -- The following list summarizes the MicroPDP-11 functional diagnostic tests.

1. Register Address Test - this test verifies that the Q-bus can read and write to the unit under test. DHU11/DHV11 mode.

2. MASTER.RESET Test - verifies that master reset clears within 5 seconds. DHU11/DHV11 mode.

3. MASTER.RESET (skip self-test) Test - verifies that the master reset clears in approximately 30 milliseconds when the skip self-test sequence is used. DHU11/DHV11 mode.

4. RX.CHARACTER Field Test - verifies that the data bits of the codes in the receive FIFO after a master reset and skip self-test are consistent with the skip self-test codes. DHU11/DHV11 mode.

5. RX.FLAG Field Test - verifies that the three data status bits (overrun, framing, and parity error bits) are all set on each of the skip self-test codes in the FIFO after a master reset and skip self-test sequence. DHU11/DHV11 mode.

6. RX.DATA.AVAIL Test - verifies that the RX.DATA.AVAIL bit is set when the skip self-test codes are in the FIFO, and that it clears after they have been read. DHU11/DHV11 mode.

7. RX.DATA.VALID Test - verifies that the RX.DATA.VALID bit is set for all the codes in the FIFO, and clear after all the codes have been read. DHU11/DHV11 mode.

8. RX.LINE FIELD Test - verifies that the RX.LINE line fields are correct for the skip self-test codes. DHU11/DHV11 mode.

9. BMP Check Test - verifies that the unit does not immediately fail the background monitor program, as this may invalidate further tests. DHU11/DHV11 mode.

10. Skip Self-test Test - verifies that the unit skips the self-test in the time allowed, and that the FIFO contains the correct codes after its completion. DHU11/DHV11 mode.

11. DIAGNOSTIC.FAIL Test - verifies that by using the skip self-test sequence, the DIAG.FAIL bit sets and clears with the allowed times.

12. Self-test Test - verifies that the unit's self-test executes within the correct time, and that the correct codes are returned in the FIFO after its completion. DHU11/DHV11 mode.

13. Self-test Fail Test - verifies that the unit will report errors when it is forced to fail by using the special fail self-test sequence (decimal 146314 written to the transmit-buffer count register). The DIAG.FAIL bit sets, and at least one self-test failure code(231) is in the receive FIFO. DHU11/DHV11 mode.

14. Chip Version Number - verifies that the chip version numbers are reported correctly, and, if requested, prints them out. DHU11/DHV11 mode.

15. CSR<4> (Skip Self-test) Test - verifies that when this bit is set (at the same time as master reset), the unit remains inactive with the master reset bit set; and when CSR bit 4 is subsequently cleared, the unit becomes active and reports six skip-self-test codes in the receive FIFO. Any BMP codes found in the FIFO are saved to be reported later. DHU11 mode only.

16. Word Access Read/Write Test - verifies that the registers respond correctly to read and write accesses. DHU11/DHV11 mode.

17. Word Access Read-Modify-Write Test - verifies that the registers will respond correctly to read-modify-write accesses. DHU11/DHV11 mode.

18. Byte Access Read/Write Test - verifies that the registers will respond correctly to byte read/write accesses. DHU11/DHV11 mode.

19. Byte Access Read-Modify-Write - verifies that the registers will respond correctly to byte read-modify-write accesses. DHU11/DHV11 mode.

20. TX.DATA Invalid Test - verifies that if a character is written to the transmit character register(TXCHAR) without TX.DATA.VALID<15> set, no TX.ACTION report occurs. DHV11 mode only.

21. TX.DATA Valid Test - verifies that if a character is written to the transmit character register (TXCHAR) with TX.DATA.VALID<15> set, a TX.ACTION occurs. DHV11 mode only.

22. TX.ENABLE (Inactive) Test - verifies that when a line's TX.ENBLE bit is clear, transmission will not take place on that line. DHU11/DHV11 mode.

23. TX.ENABLE (Active) Test - verifies that when a line's TX.ENABLE bit is set, transmission will take place on that line. DHU11/DHV11 mode.

24. DMA Start Test - verifies that each DMA.START bit will initiate a DMA transmission on a line. DHU11/DHV11 mode.

25. DMA Abort Test - verifies that each DMA.ABORT bit will stop a DMA transmission, return a TX.ACTION, and successfully restart the DMA. DHU11/DHV11 mode.

26. DMA Error Test - verifies that the DMA.ERROR bit in the CSR reports DMA errors correctly when they occur. DHU11/DHV11 mode.

27. FIFO Data Test - verifies that the FIFO will hold 256 characters without corrupting data, and that the overrun set is clear. DHU11/DHV11 mode.

28. OAUTO Inactive Test - verifies that the unit will not respond to incoming XON and XOFF characters when O.AUTO is clear. DHU11/DHV11 mode.

29. OAUTO Active Test - verifies that the unit responds correctly to incoming flow control characters when active. DHU11/DHV11 mode.

30. XON-XOFF Filtering Test - verifies that when DISAB.XRPT (LPR<0>) and OAUTO (LNCTRL<4>) are set, XON-XOFF characters are not placed in the receive silo. DHU11/DHV11 mode.

31. Interrupt Test - verifies that the unit will generate reception and transmission interrupts correctly. DHU11/DHV11 mode.

32. Diagnostic Field (BMP) Test - verifies that a request to the unit to report BMP status codes is complied within the specified time. All active lines are tested. DHU11/DHV11 mode.

33. Break Generation Test - verifies that all serial transmit lines can generate a Break by setting the BREAK bit in the associated LNCTRL register. DHU11/DHV11 mode.

34. No Overrun Error Test - verifies that the unit under test will not report data overrun errors when they do not occur. DHU11/DHV11 mode.

35. Overrun Error Test - verifies that the unit will report data overrun errors when 261 characters are received. DHU11/DHV11 mode.

36 to 45. Modem Tests - skipped for CXA16/CXB16.

44. Transmit Line Test - verifies that the transmit lines and receive lines are working correctly through the device cables, distribution panel, and loopback connectors. Executed only if external mode is selected. DHU11/DHV11 mode.

45. Transmit Lines Interaction Test - looks for any interaction between lines. Executed only if one of the external loopbacks is selected. DHU11/DHV11 mode.

46. RXTIMER Test - verifies that the hold-off timer for receive interrupts is operating correctly, and that the three-quarters full level overrides the timer. DHU mode only.

47. Modem Test - skipped for CXA16/CXB16.

48. Transmit FIFO Test - verifies that the FIFO will hold 64 unique characters, and that only one interrupt occurs for all 64 characters. DHU11 mode only.

49. DMA Address Test - verifies that the unit can access the full memory which is on the machine through DMA access. DHU11/DHV11 mode.

51. Keyboard Echo Test - allows the operator to test terminal links (or other communications links), which are attached to unit serial ports, from remote ends of the links. DHU11/DHV11 mode.

52. Single Character Test - verifies that the unit will transmit and receive correctly using non-DMA mode at various line parameters. DHU11/DHV11 mode.

53. DMA Mode Test - verifies that the unit will transmit and receive correctly using DMA at various line parameters. DHU11/DHV11 mode.

54. Framing Error Test - verifies that forced framing errors are reported correctly. H3101 loopback only. DHU11/DHV11 mode.

55. Parity Error Test - verifies that forced parity errors are reported correctly. H3101 loopback only. DHU11/DHV11 mode.

56. Split Speed Test - verifies that the unit will function correctly using different transmit and receive speeds on each active line. H3101 loopback only. DHU11/DHV11 mode.

57. Report BMP Codes Test - this pseudo-test reports the first 32 characters which were discovered in the FIFO during the execution of the other tests. This avoids interruption of the other tests by these codes if they are not critical to the performance of the tests. DHU11/DHV11 mode.

4.6.1.2 Loopback Connectors -- Two loopback connectors are available for the CXA16/CXB16:

- H3101: Eight-line loopback (loops 0 to 0, 1 to 1 ...)
- H3103: Single-line loopback.

4.6.2 DECX11 Object Module
The CXA16/CXB16 Object Module is the same as for the DHV11: XDHV**.OBJ.

NOTE

Early versions of this module DO NOT support the CXA16/CXB16.

The minimum parameters which must be specified are:

1. DVA (Device Address)
2. VCT (Vector Address)
3. DVC (Device Count), if more than one.

The default operating mode is with all lines looped back internally at 9600 baud.

4.7 MicroVAX II SERVICE MODE DIAGNOSTICS

The CXA16/CXB16 uses the same diagnostic module as for the DHV11, NADHA*.

4.7.1 Configuration Tests

The module is 'sized' to check the number of lines (8 or 16), and whether it supports modems or is data-leads-only. This information, together with the control chip and OCTART revision levels, is added to the system configuration file.

4.7.2 Field Service Functional Tests

Before the diagnostic runs a field service test, the setup procedure for the CXA16/CXB16 diagnostic is executed. You will be prompted to attach any loopback connectors or cables (for instance, bulkhead loopback connectors). On pressing the RETURN key, all ports are sized to see to which ports (on the same controller) the loopbacks are attached. This information is then displayed, and the field service test are started. This setup procedure is run only once after configuration of the diagnostic system.

TEST 1: In this test, the CXA16/CXB16 is initialized and addressability checks are made on various registers.

TEST 2: This test uses loopback connectors and cables sized during the setup procedure to send single characters in programmed output mode over port 0, using baud rates of 300, 1200, 9600, and 38.4k bauds, and with various data lengths, stop bit sizes, and parity options. It should be noted that, due to execution time constraints, this is the only test, other than the utilities, which exercises the entire range of character options.

TEST 3: This test performs extensive DMA testing through loopback connectors and cables sized during the setup procedure, using different buffer sizes. All ports are tested simultaneously at 19.2Kbytes/s. FIFO overrun and DMA abort are tested. In addition, if a port is determined to be connected to a port other than itself, a more thorough test of outgoing data flow-control is performed.

TEST 4: This test uses internal loopback connectors and cables sized during the set-up procedure to perform extensive flow-control testing. OAUTO, IAUTO, and FORCE.XOFF functions are exercised. The break signal function is also tested.

TEST 5: (DHU11 mode only) This test uses loopback connectors and cables sized during the setup procedure to loop back the data from the transmit FIFO to the receiving channels, to verify that the transmit FIFO buffer can hold 64 characters without corrupting the data.

4.7.3 Field Service Exerciser Tests

The Field Service exerciser performs each of the functional service tests 2 to 5, using loopback connectors and cables found during setup. All applicable ports are tested, but only at 19.2Kbytes/s.

4.7.4 Utilities

The three utilities provide simple routines to run echo tests and terminal tests on specific lines. They can be used to locate cable faults, using the loopback connectors to loop data back to the module, or to check that a terminal, or remote equipment, is correctly transmitting and receiving characters.

Utility 1 This utility permits staged testing of the CXA16/CXB16 and associated cables/connectors. The diagnostic requests configuration parameters from the operator, and then repeatedly tests the port(s) for data loopback integrity. The operator may, by strategically placing loopback connectors at various points in the communications path, isolate defective units. The operator is not prompted as to where to place the loopback connectors, nor does the diagnostic interpret the results of the tests.

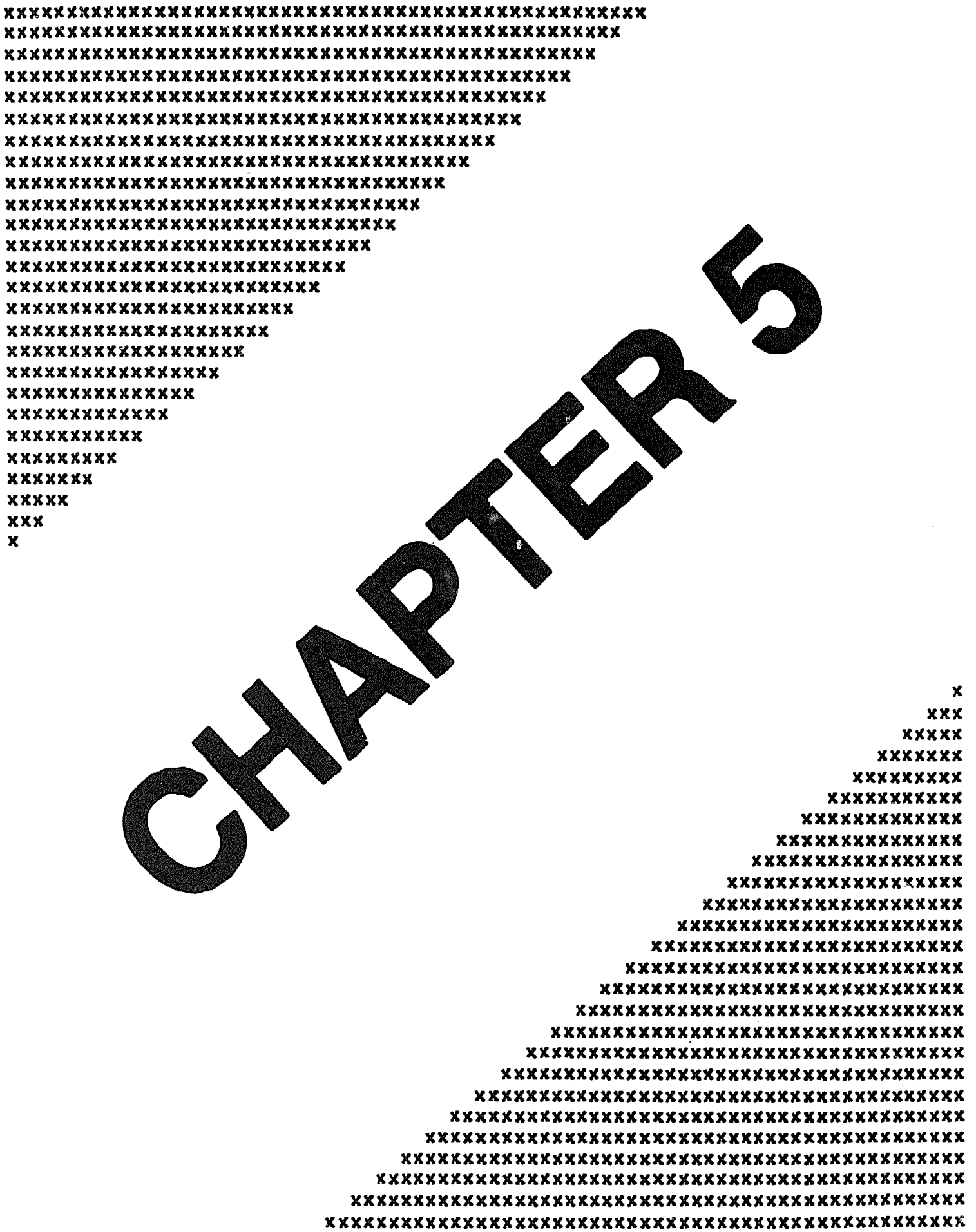
Utility 2 This utility permits testing of remote communications devices. The diagnostic puts all ports into remote loopback maintenance mode. All characters sent to any port will be echoed back to the sender, at baud rates up to 9600 baud. No operator configuration is necessary. Because of inherent CXA16/CXB16 buffering limitations, characters may be lost if transmitted too rapidly by the terminal.

Utility 3 This utility sends test strings to the port(s) specified by the operator to aid diagnoses of remote communications devices. The operator may choose any baud rate or character specifics. The port under test is configured with auto-flow control for outgoing characters.

4.8 FIELD-REPLACEABLE UNITS (FRUs)

The FRUs are:

Reference No.	Item
M3118-YA	CXA16 module
M3118-YB	CXB16 module
BC16D-25	36-wire cable
H3104	Cable concentrator


CHAPTER 5

CHAPTER 5
TECHNICAL DESCRIPTION

5.1 SCOPE

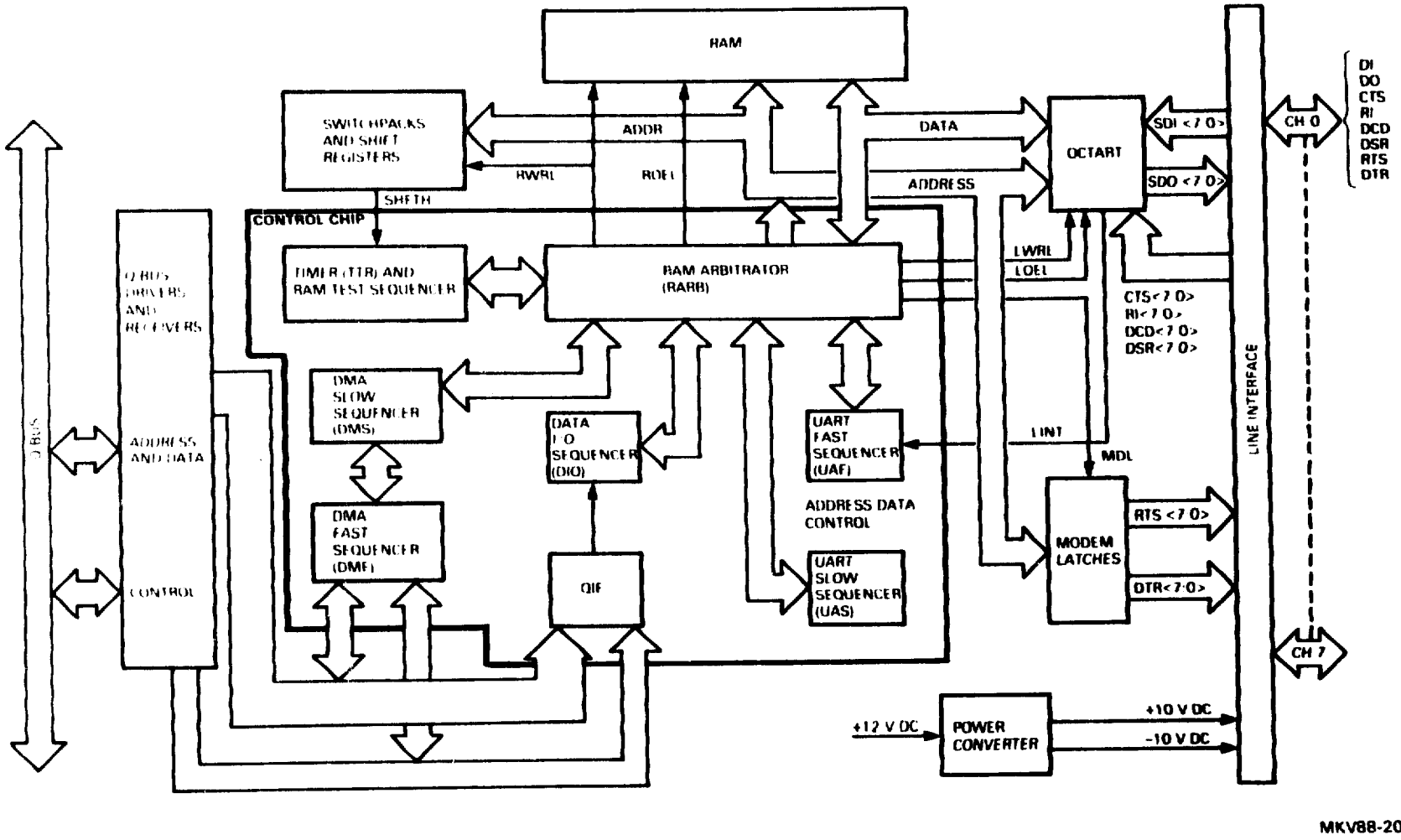
This chapter gives a technical description of the CXA16/CXB16 asynchronous multiplexer. Figure 5-1 shows a block diagram of the CXA16/CXB16, and each block identified in the diagram is described in this chapter. While reading this chapter, you may find it useful to have access to a CXA16/CXB16 printset (part number MP-02381).

.2 OVERVIEW

Figure 5-1 shows a block diagram of the CXA16/CXB16. It can be broken down into the following sections.

- RAM -- stores control information and provides the data buffers
- OCTARTs -- each OCTART contains eight asynchronous receiver/transmitters, baud-rate generators, and interface logic
- Control Chip -- contains sequencers and other logic to implement all the functions of the CXA16/CXB16 outside of the OCTART.
- Switchpack and Shift Register Section -- defines the device address and vector, and the mode of operation
- Q-bus Drivers and Receivers -- provide the electrical interface to the Q-bus
- Line Interface -- converts signals to and from line levels
- Power Converter -- provides positive and negative 10 V supplies for the line interface (CXA16 only).

Each of these blocks is described in more detail in Section 5.3. Section 5.4 describes the flow of data through the module.



MKV88-2055

Figure 5-1 CXA16/CXB16 Block Diagram

5.3 CXA16/CXB16 FUNCTIONAL BLOCKS

5.3.1 RAM

The way in which the RAM is assigned to the various functions is shown in Figure 5-2. Two Kbytes of RAM are used. This RAM is divided into three main sections:

- The top 1 Kbyte contains the 64-character transmit FIFOs, one per channel.
- The next 512 bytes contain the 256-word receive FIFO. The low byte of each word contains a received character, and the high byte contains status information, such as channel number and error status.
- The bottom 512 bytes contain the 32-byte control register areas, one per channel. These registers control the data flow through the CXA16/CXB16, through the sequencers in the control chip.

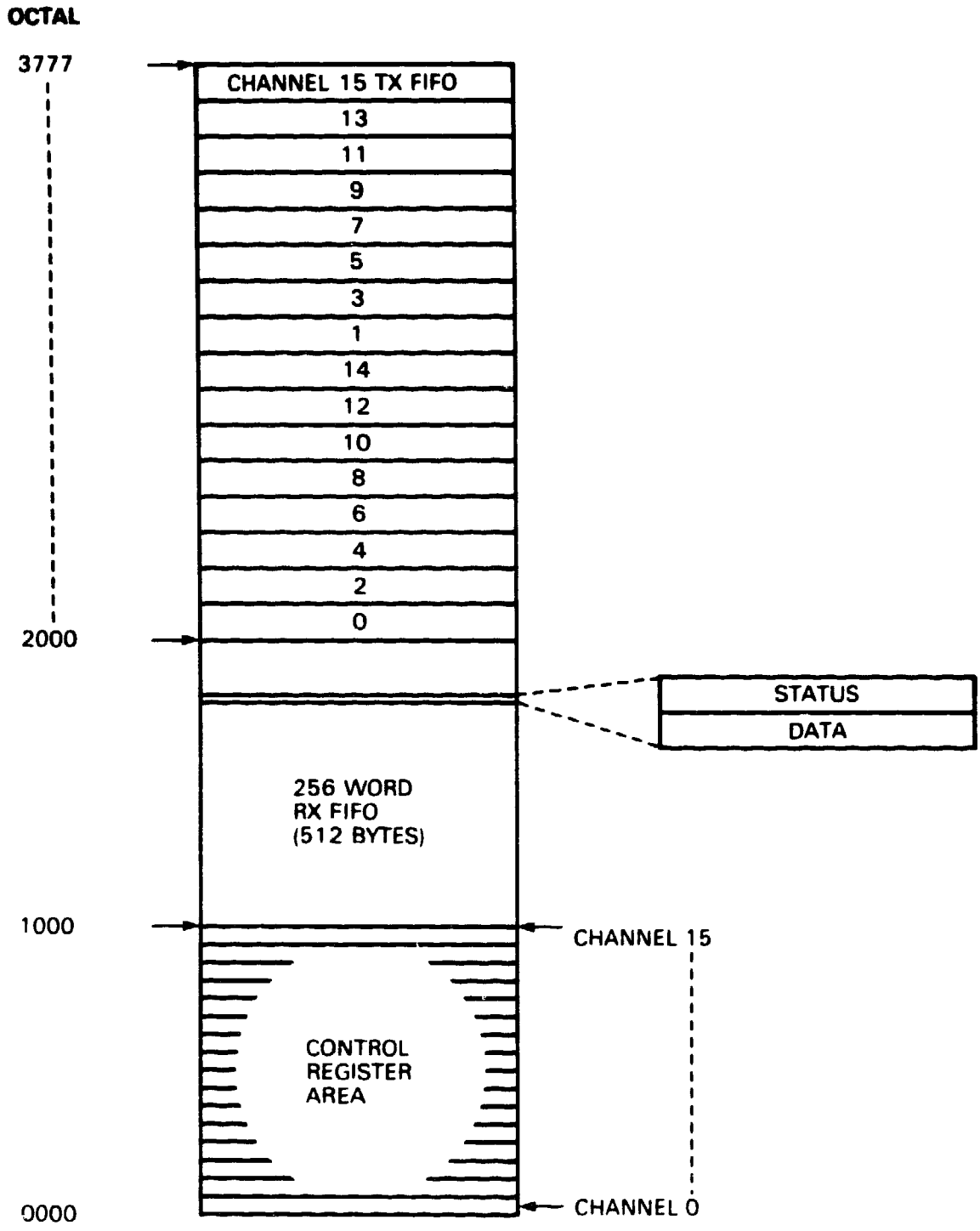


Figure 5-2 RAM Assignment

5.3.2 OCTART

Each OCTART contains 8 asynchronous receiver/transmitters, a sixteen-output baud-rate generator and interface logic. They are compatible from a serial-line viewpoint with those in the 2681 DUART. Each channel has only a single buffer on both transmit and receive. The UART fast sequencer on the control chip (see Section 5.3.3.6), together with the RAM, provides double buffering for the transmitters and a 4-word-per-line FIFO for each receiver, thus completing compatibility with the 2681 DUART.

The OCTART is shown in greater detail in Figure 5-3.

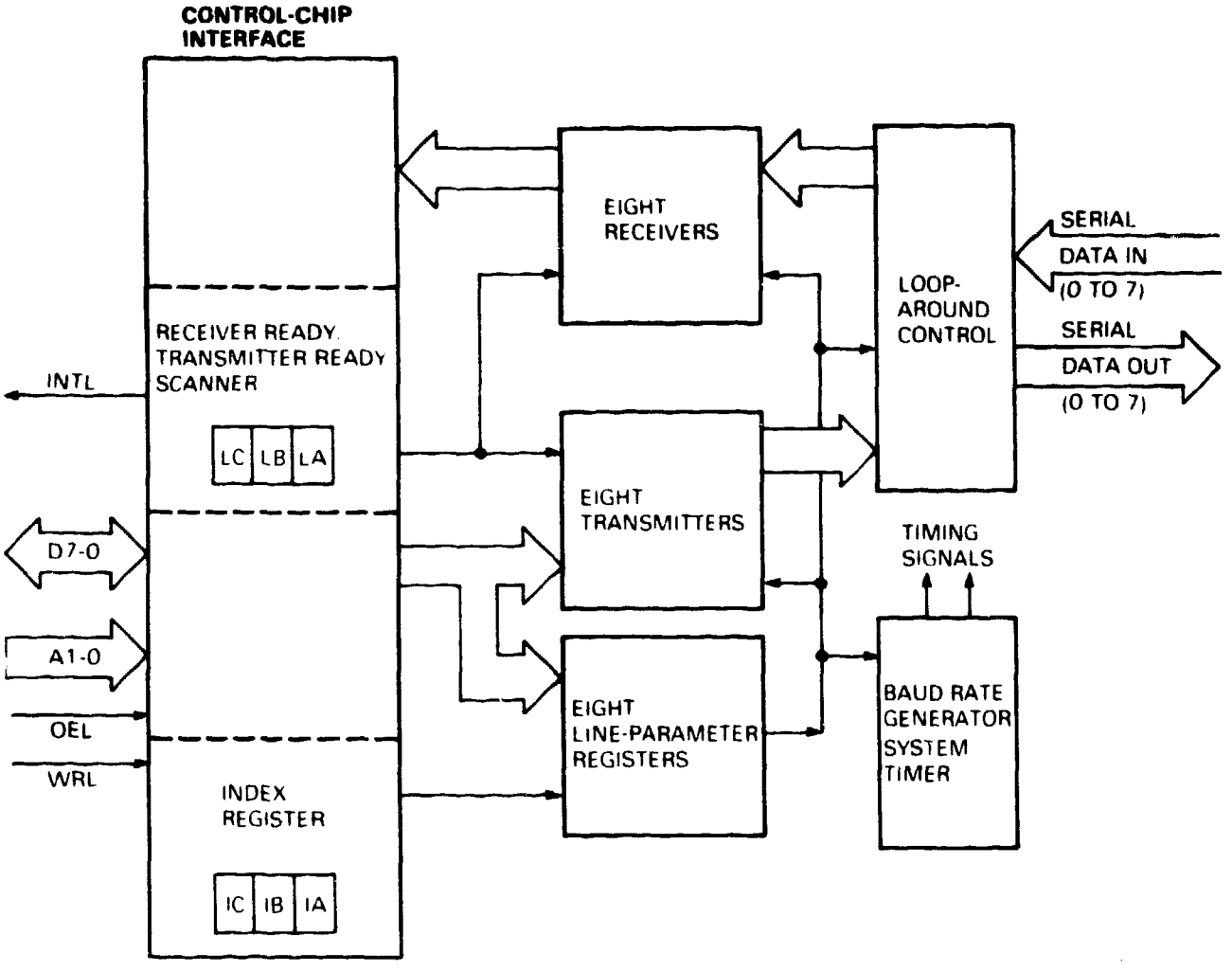


Figure 5-3 OCTART

5.3.2.1 Control-Chip Interface -- This manages two separate sources of information for the OCTARTs:

- Receiver/Transmitter-Ready Scanner Register bits L<C:A>, together with the receive ready bit
- Index Register bits I<C:A>.

The scanner logic scans each channel in turn until it locates a channel needing servicing. It then asserts the interrupt line to the control chip. A channel is ready for servicing when either it is free to accept a character for transmission, or a character has been received, as indicated by the receiver-ready bit. During the servicing of this interrupt by the control chip, the L<C:A> bits tell the control chip which channel has interrupted. If it is a receive interrupt, the received character is presented on the D<7:0> lines. If it is a transmit interrupt, the control chip will either present a character on D<7:0>, or performs a dummy transfer. Note that the receivers are given priority over transmitters.

Register bits I<C:A> select one of eight channels during a register access. Address lines A<1:0> select one of four registers associated with that channel.

5.3.2.2 Line-Parameter Register -- Each line has its own line-parameter register derived from the Q-bus-addressable LPR and LNCTRL registers.

5.3.2.3 Loop-Around Control -- This implements the functions of the maintenance mode bits (see Section 3.2.2.9).

5.3.2.4 Baud-Rate Generator -- This provides 16 independent baud rates for transmit and receive. Transmitter baud rates are independent from the receive rates, and there is no interdependency between channels.

5.3.3 Control Chip

The control chip contains all the necessary logic and sequencers to control the operation of the CXA16/CXB16. The functions of the control chip can be broken down into seven blocks:

- Timer and RAM test sequencer (TTR)
- Q-bus interface (QIF)
- DMA slow sequencer (DMS)
- DMA fast sequencer (DMF)
- Data I/O sequencer (DIO)
- UART fast sequencer (UAF)
- UART slow sequencer (UAS)
- RAM arbitrator (RARB).

5.3.3.1 Timer and RAM Test Sequencer (TTR) -- This sequencer controls the power-up sequence of the module. On power-up, the TTR:

- Resets the OCTARTs
- Clocks the switchpack settings into the control chip. RAM address lines are used to select the shift register operation, and RWR.L (RAM write) is used to clock the shift register (see Section 5.3.4 for more information)
- Tests the RAM (which takes 26 ms) and leaves the RAM in a defined state, which reflects the power-up state described in Section 3.3.1.

The sequencer also generates internal timing signals for the control chip.

5.3.3.2 Q-bus Interface (QIF) -- This block contains the logic to handle data I/O and interrupt operations.

The QIF also includes other functions, such as:

- A special DOUT handler, so that the Q-bus is given a fast response time without having to wait for RAM access
- Logic to handle interrupt request generation and vector assertion (using data read from the switches at power-up).

Some device register bits are also implemented in this block, such as:

- Indirect address register pointers
- Interrupt enable bits.

5.3.3.3 DMA Slow Sequencer (DMS) -- DMA operations are controlled by a combination of the DMF (DMA-Fast) sequencer in this block and also the DMS (DMA-Slow) sequencer (described in Section 5.3.3.3).

This sequencer has three independent functions.

- To handle a DMA transfer by controlling the DMF sequencer
- To handle an ABORT request
- To report a TX.ACTION at the end of a DMA operation (this includes successful operations, aborted operations, and operations terminated after a bus error)
- To scan the channels until it detects the need for a DMA operation.

5.3.3.4 DMA Fast Sequencer (DMF) -- The function of the DMF sequencer is to become Q-bus master, and then acquire characters from the Q-bus at maximum Q-bus speeds. At an average rate of 1 word every 850 ns in block-mode DMA, the DMF acquires up to 8 words (16 characters) during its bus mastership. In non-block operations, up to 4 words are acquired while it is bus master. These characters are transferred to the transmit FIFO under the control of the DMS sequencer. The DMF sequencer operates as a slave to the DMS sequencer.

5.3.3.5 Data I/O Sequencer (DIO) -- This sequencer waits in an idle loop until signals from the Q-bus (through the QIF) cause it to do data I/O operations -- DATI, DATO(B), and DATIO(B).

This sequencer also detects operations on certain registers and takes appropriate action. For example, after a write operation to a LPR or LNCTRL register, it sets a bit in the RAM control area for that channel to indicate to the UAS sequencer that the contents may have changed.

5.3.3.6 UART Fast Sequencer (UAF) -- This sequencer handles interrupts from the OCTARTs. When an interrupt is received, the UAF reads the OCTART status register to find the channel number, and whether it is a receive or transmit interrupt.

For a transmit interrupt, having checked that there is a character ready to transmit, the UAF transfers the character directly from the RAM to the OCTART. For a receive interrupt, the UAF transfers the character directly from the OCTART to the top of the 4-word FIFO in RAM (low byte). Then the error status, which had previously been read from the OCTART status register, is written to the FIFO (high byte).

5.3.3.7 UART Slow Sequencer (UAS) -- This sequencer scans all 16 channels (in 160 microseconds, to guarantee operation on all channels at maximum possible data rates). For each channel, the UAS:

1. Moves received characters from the bottom of the 4-word FIFO in RAM into the 256-word shared receive FIFO. It also recognizes and acts upon received XON/XOFF control characters.

2. Implements one of the following five actions (listed in priority order) before repeating step 1 for the next channel:

   - Line-parameter register written check -- it checks whether this register has been changed by the DIO sequencer
   - Line-parameter register change -- if during the previous pass it detected that this register had been changed, it now updates the OCTART registers with the new information
   - BMP Report -- generates a Background Monitor Program report
   - Transmits a single character from the TXCHAR register, or transmits XON or XOFF characters
   - Transmits from the transmit FIFO.

The UAS block also includes:

- Receive FIFO control
- TX.ACTION FIFO control
- RX.TIMER logic (only used in DHU11 programming mode)
- Background Monitor Program (BMP) logic.

5.3.3.8 RAM Arbitrator (RAPB) -- This arbitrates between the six RAM access sequencers described in sections 5.3.3.1 and 5.3.3.3 to 5.3.3.7, giving one of them access to the RAM/OCTART data bus, depending on the priority scheme described here.

The TTR sequencer has the top priority so that it gets 100% RAM access for the first 26 ms after power-up. This guarantees that the RAM will be powered-up to an operational state before any other sequencer can begin. After 26 ms this sequencer idles.

Although it cannot gain access to RAM during the 26 ms power-up time, the DIO sequencer still operates. This is needed for CSR access and SKIP SELF-TEST operations from the Q-bus (see Section 3.2.2.1).

To ensure that the progress of each sequencer is predictable, RAM access is alternated between the two UART sequencers (UAF and UAS) and the two CPU sequencers (DIO and DMS). During 'CPU-time' only CPU sequencers are granted access. During 'UART-time', UART sequencers have priority, but if neither UART sequencer requires access, a CPU sequencer can steal the cycle. The overall effect of this is that:

- The UART sequencers progress at a predictable rate. This guarantees service to each channel at a rate fast enough to maintain maximum throughput while still allowing a required minimum time between OCTART write operations

- Access by the CPU sequencers is maximized (and hence Q-bus BRPLY delay times minimized) by allowing them RAM access during unused UART cycles, along with normal CPU cycles.

In 'UART-time', the UAF has priority over the UAS to make sure all data transfers to and from the OCTART are done as quickly as possible. Of the CPU sequencers, the DIO has priority over the DMS to get the minimum delay in asserting BRPLY.

5.3.4 Switchpack and Shift Registers

The TTR sequencer shifts-in the contents of the shift registers after the module has been reset (by BINIT or MASTER.RESET). Since the switchpack contents are only examined at this time, the module must be reset (BINIT or MASTER.RESET) in order to set new values.

The TTR sequencer in the control chip has been designed for use in several different applications, and therefore needs to determine whether the module has 8 or 16 channels, and whether it is a Q-bus or UNIBUS application (the CXA16/CXB16 is set up for 16 channels on Q-bus). Figure 5-4 shows a simplified diagram of the switchpack and shift register logic.

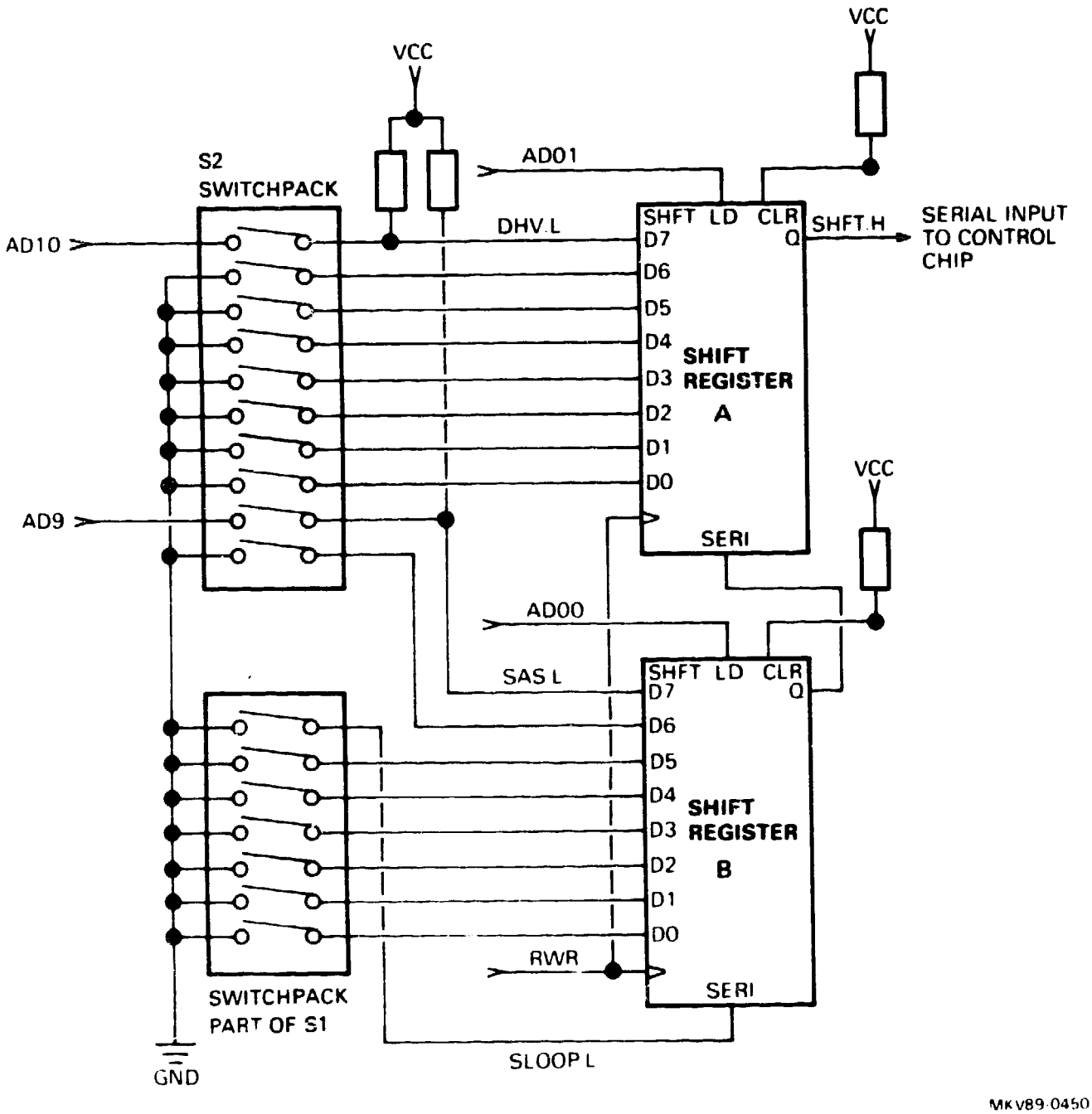


Figure 5-4 Switchpack and Shift Register Logic

First, the switch settings connected to the D<7:0> inputs of the shift registers are synchronously loaded into both shift registers. This is done by holding AD00 and AD01 low while pulsing RWR.L. The switch settings are clocked into both shift registers by the trailing edge of RWR.L. AD09 and AD10 are held low during this first operation to ensure that the respective D7 settings (DHV.L and SA5.L) are loaded in the same way as all the other switches.

After this load, the first bit (D7 of shift register A) is present at the SHFT.H input to the control chip, and is latched internally by the TTR sequencer. Then 16 RWR.L pulses are applied with AD00 and AD01 high (this selects the shift function). The remaining bits in the shift register are thus clocked into the control chip and latched by the TTR, including the last shifted-in setting (SERI input) to the B shift register (SLOOP.L), making a total of 17 bits.

Two further settings need to be determined.

- Whether the device supports 16 lines (as does CXA16/CXB16) or 8 lines
- Whether the device operates on the Q-bus (as does CXA16/CXB16) or on the UNIBUS.

After the 17 switch settings have been latched, the control chip sets AD09 and AD10 high. It then selects the load function (AD00 and AD01 low) and pulses RWR.L. Because of the high level on AD09 and AD10, a high level is latched into D7 of both shift registers, regardless of the switch settings.

The control chip now tries to clear these two bits. If the device supports 16 lines, AD08 is connected to the CLR input of shift register A. If the device operates on the UNIBUS, AD02 is connected to the CLR input of shift register B. The control chip pulses AD02 and AD08 low to try to clear the registers. On the CXA16/CXB16, neither of these connections is made (both CLR inputs are simply pulled high); therefore the registers do not clear. The control chip can now test the state of the SHFT.H line and determine (since SHFT.H is high) that the CXA16/CXB16 has sixteen lines. After applying eight RWR.L pulses, a second test of the SHFT.H line shows that the CXA16/CXB16 operates on Q-bus (SHFT.H is again high).

5.3.5 Q-bus Drivers and Receivers

Six DC021 Q-bus driver/receiver chips are used. The direction (transmit or receive) of each is controlled by signals from the control chip. Bus contention is avoided while the direction is being changed by disabling the transceivers, using enable inputs EN1.L and EN2.L.

5.3.6 Line Interfaces

The interface to the serial lines on the CXA16 is provided by two octal line receiver chips and two octal line driver chips, and on the CXB16 by two octal line receiver chips and four quad line drivers. These are inverting buffers which convert between line levels at the J1 and J2 connectors, and TTL levels at the OCTART.

5.3.7 Power Converter (CXA16 only)

The CXA16 line drivers require + and - 10 V dc. The Q-bus backplane supplies only +12 V dc. A nominal +10 V is generated by dropping down the +12 V across two diodes. The -10 V is derived from +12 V by a voltage converter. This device uses switch-mode power-supply techniques to generate the negative voltage. The circuit is built around a TL494 switching regulator (see Figure 5-5). This uses pulse-width modulation to regulate the -10 V output.

Switching pulses from the modulator switch a transistor (Q1) to convert a dc input to a pulsed dc current in an inductor. When Q1 is switched on, point X becomes positive, causing current to flow through L. During this period energy is stored in the inductor. When Q1 is switched off, the polarity of the voltage across the inductor is inverted, and the energy stored in the inductor is transferred via the now forward-biased diode to the smoothing capacitors. As current is transferred to the output, the voltage at X rises until the diode is cut off again. The circuit will stay in this state until the next switching pulse turns Q1 on.

The inset of Figure 5-5 shows typical inductor current waveforms. With Q1 switched on, current rises linearly until Q1 is switched off again. With Q1 switched off, the collapsing field in the inductor maintains current flow which reduces linearly as it is transferred to the output. The wider the switching pulses, the more the power transferred to the output.

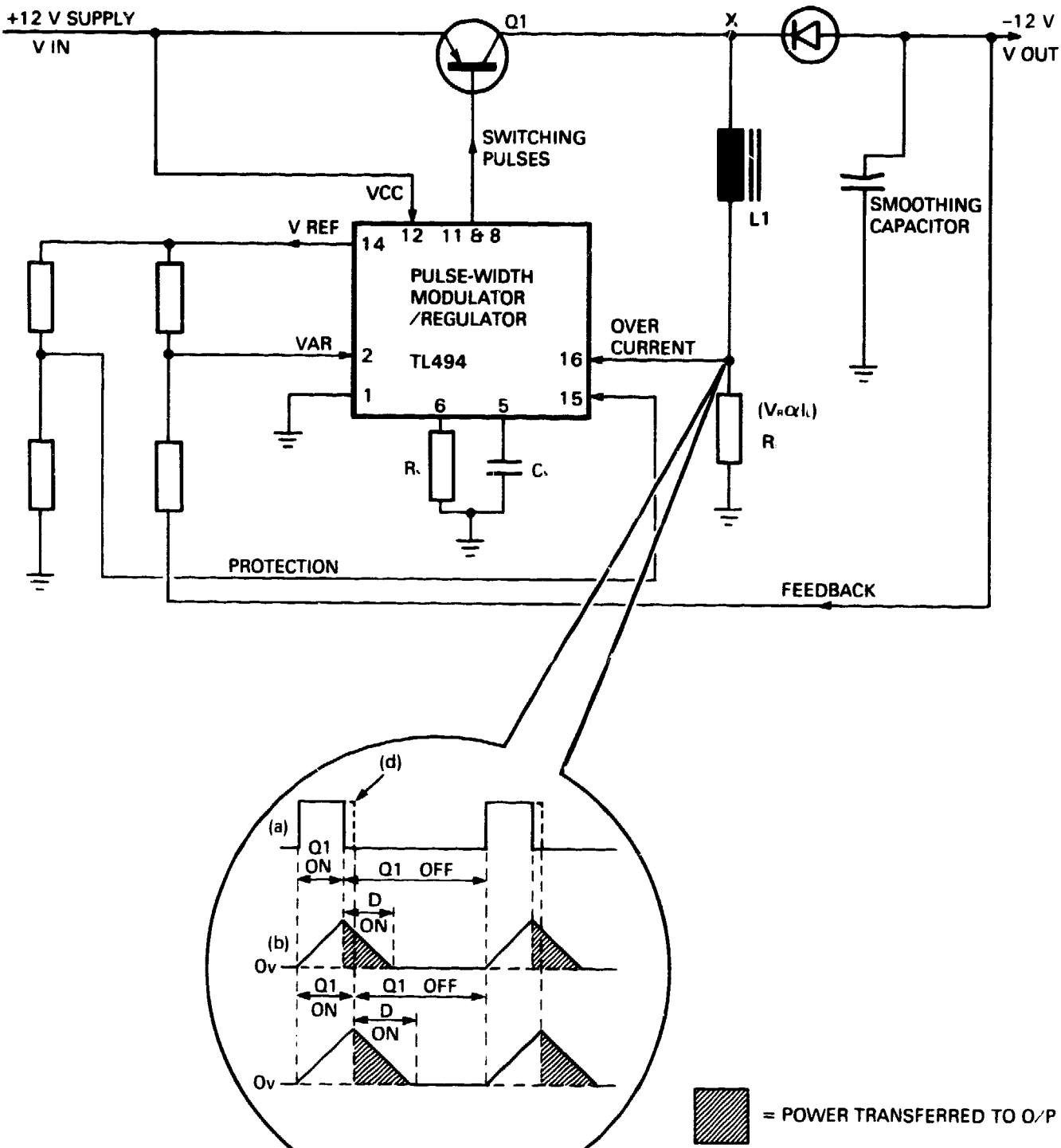


Figure 5-5 Power Converter

Feedback from the output is compared with a reference voltage. If the output is too high, the pulse width is reduced; if it is too low, the pulse width is increased. This feedback action maintains output voltage regulation for varying loads. This same method of comparison is used to implement over-current protection. The inductor current is sensed across Rc. If this exceeds a preset limit, the switching pulse width is reduced. The switching frequency (60 kHz) is selected by Rs and Cs. If the oscillator is working, a sawtooth waveform can be seen at pin 5.

5.4 DATA FLOW

This section describes the general flow of data through the CXA16/CXB16 between the Q-bus and the serial lines, for both the receive and the transmit operations.

5.4.1 Data Flow for Character Reception

Data is clocked into the OCTART in serial form. When the OCTART has completed the serial-to-parallel conversion, it generates an interrupt (LINT.L or HINT.L) to the control chip (see Figure 5-6). The UAF sequencer quickly transfers the data into the four-word FIFO for the appropriate channel in RAM, and sets a receive-data-available flag. The UAS sequencer sees this flag and processes the character by moving it, together with the channel number and error information, to the top of the 256-word shared receive FIFO.

The CPU accesses the bottom of the receive FIFO by reading the RBUF register. This transfer is handled by the DIO sequencer.

5.4.2 Data Flow for Character Transmission

Characters to be transmitted are handled either by DMA or normal I/O operations (see Figure 5-7).

5.4.2.1 DMA Operation -- The CPU starts a DMA operation by writing the start address for the channel into TBUFFAD1 and TBUFFAD2, writing the number of characters to be transmitted into TBUFFCNT, and then setting the DMA.START bit. The DMS sequencer detects DMA.START being set and begins the DMA transfer to the appropriate transmit FIFO with the aid of the DMF sequencer. The UAS sequencer reads the status of that channel's transmit FIFO, removes the character (assuming there is one) from the bottom of the transmit FIFO, places it in the Transmit Holding Register (THR), and sets the transmit-character-available flag for the channel. The THR is contained in RAM as an element of the control register area.

The OCTART interrupts the control chip when it is able to handle a transmit character. The UAF sequencer reads the transmit-character-available flag and, if it is set, transfers the character from the THR to the OCTART. The OCTART converts it to serial format, and transmits it to the line drivers.

5.4.2.2 Programmed I/O Operation -- There are two modes of programmed I/O operation: DHU11 and DHV11.

In DHU11 mode the CPU first reads the FIFOSIZE register for the appropriate channel to find whether the transmit FIFO has space for transmit characters. If there is space, the CPU writes a word (two characters) or a byte (one character) directly into the FIFODATA register. If the CPU has further characters to transmit, it can continue to write successive characters to the FIFODATA register without having to read the FIFOSIZE register up to the number previously read from the FIFOSIZE register. This transfer is handled by the DIO sequencer, which writes the characters to the top of the transmit FIFO for the particular channel. Characters are processed by the UAS and UAF sequencers, as described for the DMA transfers above.

In DHV11 mode the host writes a single character to the TXCHAR register and sets the TX.VALID bit. This transfer is handled by the DIO sequencer. The UAS sequencer removes this character from the TXCHAR register, places it in the THR, clears the TVD bit, and sets TX.ACTION. Transfer to the serial line then continues as described above for DMA transfers.

Subsequent TXCHAR transfers must only be initiated after the TX.ACTION has been set for the previous transfer.

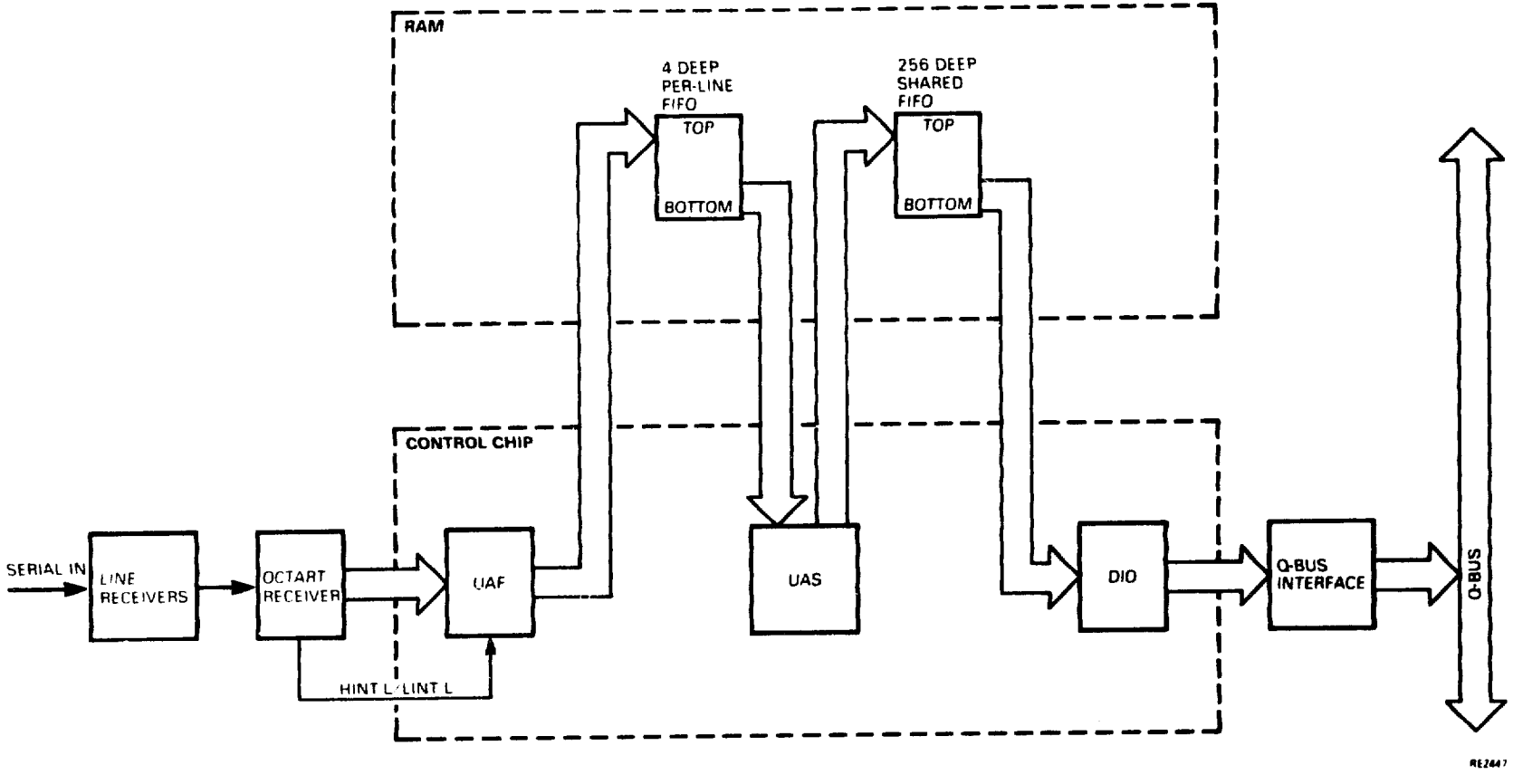


Figure 5-6 Receive Character Data Flow

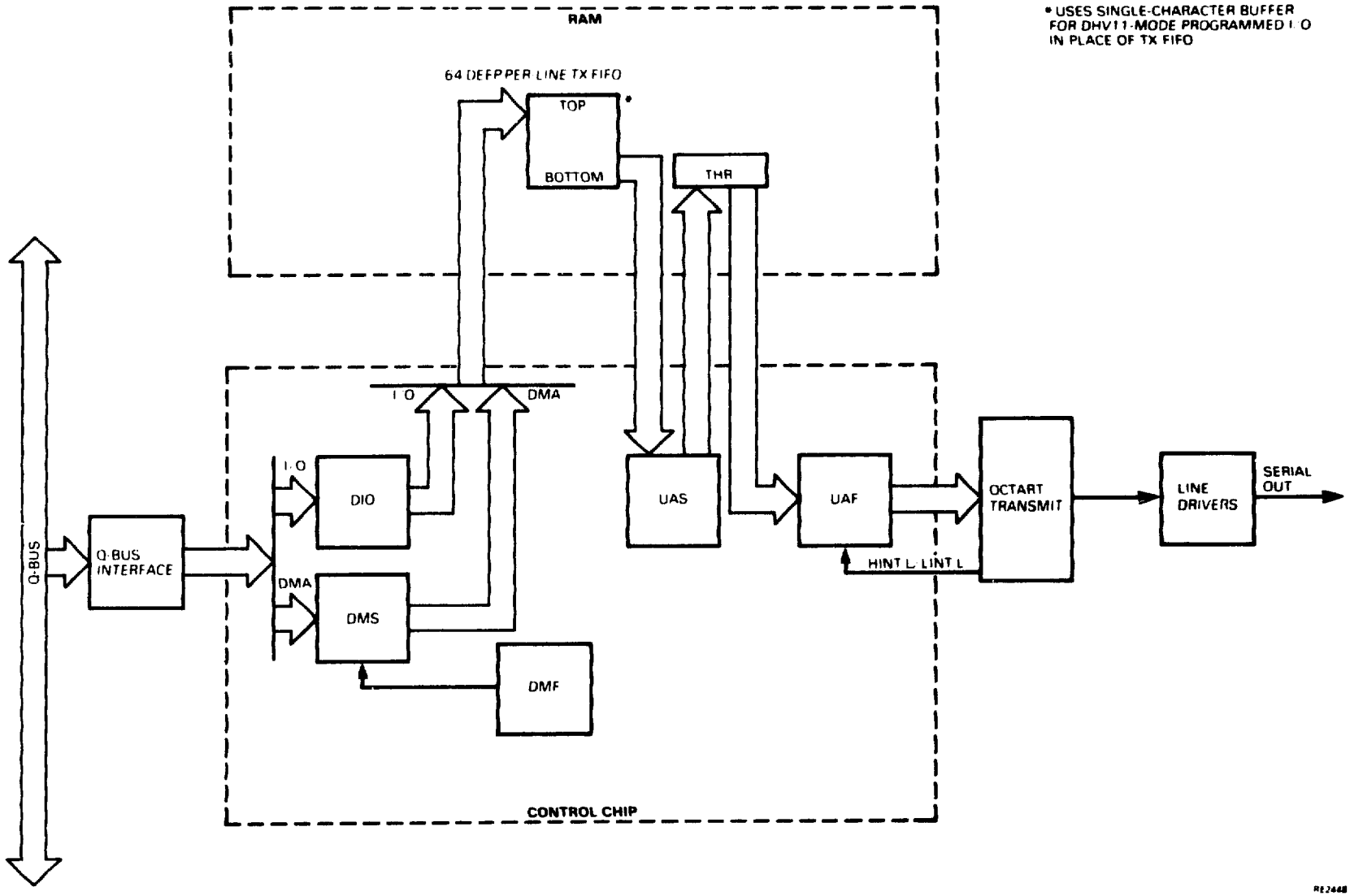


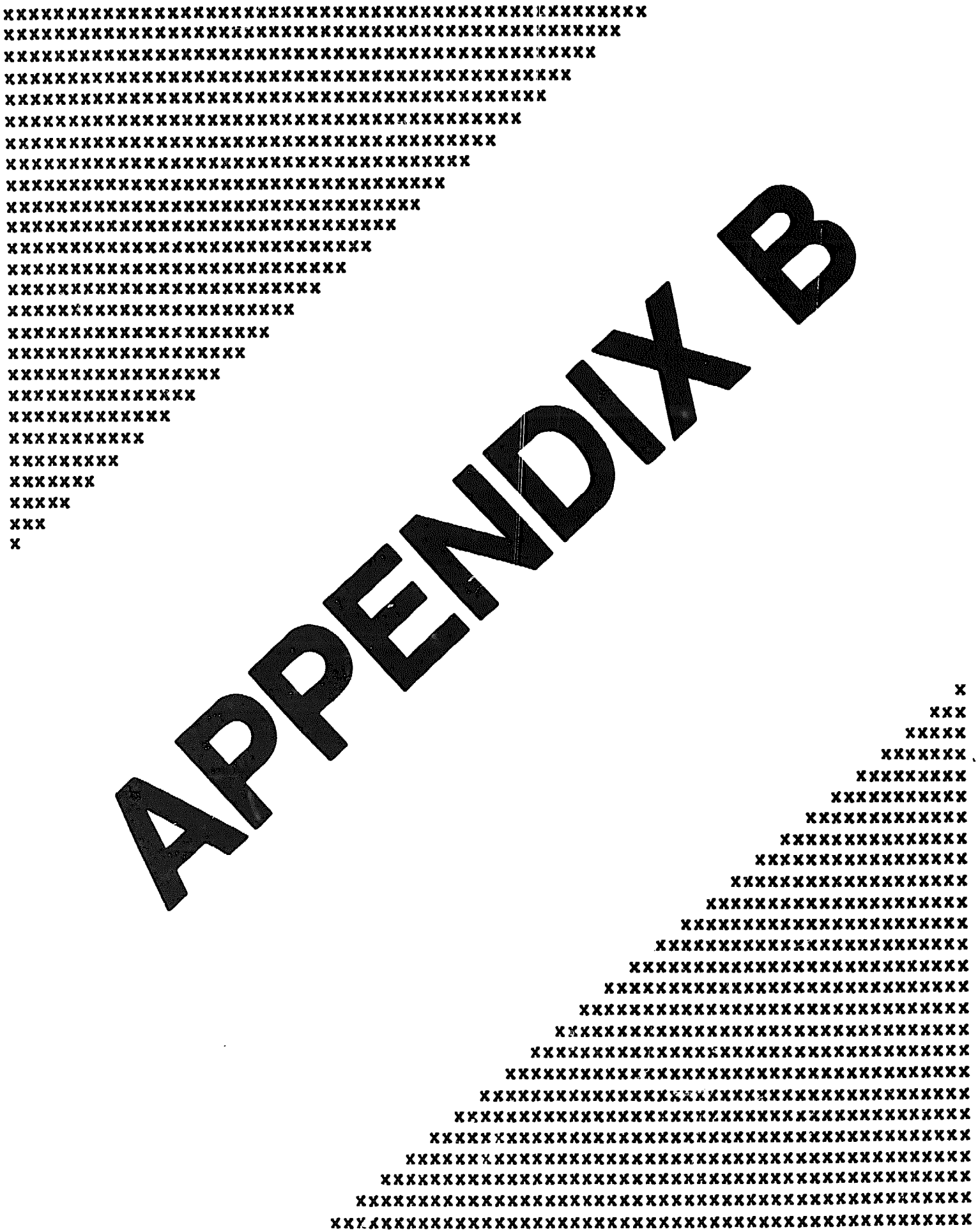
Figure 5-7 Transmit Character Data Flow


APPENDIX A

APPENDIX A
CXA16/CXB16 Q-bus CONNECTIONS

Table A-1 CXA16/CXB16 Q-bus Connections

Category	Signal	Function	Pin Number
Data/Address	BDAL0.L -- 1.L	Data/Address Lines	AU2 -- AV2
	BDAL2.L -- 15.L		BE2 -- BV2
	BDAL16.L -- 17.L		AC1 -- AD1
	BDAL18.L -- 21.L		BC1 -- BF1
Data Control	BDOUT.L	Data Output Strobe	AE2
	BRPLY.L	Reply Handshake	AF2
	BDIN.L	Data Input Strobe	AH2
	BSYNC.L	Synchronize Strobe	AJ2
	BWTBT.L	Write Byte Control	AK2
	BBS7.L	I/O Page Select	AP2
Interrupt Control	BIRQ.L	Int. Req. Level 4	AL2
	BIAKI.L	Int. Ack. Input	AM2
	BIAKO.L	Int. Ack. Output	AN2
DMA Control	BDMR.L	DMA Request	AN1
	BDMGI.L	DMA Grant Input	AR2
	BDMGO.L	DMA Grant Output	AS2
	BSACK.L	Bus Grant Acknowledge	BN1
	BREF.L	Refresh and Block Mode	AR1
System Control	BINIT.L	Initialization Strobe	AT2
Power Supplies	+5 V	Dc volts	AA2 -- DA2
	+12 V	Dc volts	AD2, BD2
Grounds	GND	Ground Connections	AC2 -- DC2
	GND	Ground Connections	AT1 -- DT1
	GND	Ground Connections	AJ1 -- BJ1
	GND	Ground Connections	AM1 -- BM1


APPENDIX B

APPENDIX B
FLOATING ADDRESSES

B.1 FLOATING DEVICE ADDRESSES

On Q-bus systems a block of addresses in the top 4K words of address space is reserved for options with floating device addresses. This range is from 17760010(octal) to 17763776(octal).

Options which can be assigned floating device addresses are listed in Table B-1. This table gives the sequence of addresses for both UNIBUS and Q-bus options. For example, the address sequences could be:

DJ11
DH11
DQ11
DU11/DUV11

Having one list allows us to use one set of configuration rules and one configuration program.

Table B-1 Floating Device Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)	Address
1	DJ11 gap	4	10	17760010
2	DH11 gap	8	20	17760020
3	DQ11 gap	4	10	17760030
4	DU11, DUV11 gap	4	10	17760040
5	DUP11 gap	4	10	17760050
6	LK11A gap	4	10	17760060
7	DMC11/DMR11 gap	4	10	17760070
8	DZ11/DZV11/ DZS11/DZ32 gap	4	10 *	17760100
9	KMC11 gap	4	10	17760110
10	LPP11 gap	4	10	17760120

* The DZ11-E and DZ11-F are treated as two DZ11s.

Table B-1 Floating Device Address Assignments (Cont.)

Rank	Device	Size (Decimal)	Modulus (Octal)	Address
11	VMV21 gap	4	10	17760130
12	VMV31 gap	8	20	17760140
13	DWR70 gap	4	10	17760150
14	RL11, RLV11 gap	4	10 *	17760160
15	LPA11-K gap	8	20 *	17760200
16	KW11-C gap	4	10	17760210
17	VSV21 gap	4	10	17760220
18	RX11/RX211/ RXV11/RXV21 gap	4	10 *	17760230
19	DR11-W gap	4	10	17760240
20	DR11-B gap	4	10 **	17760250
21	DMP11 gap	4	10	17760260
22	DPV11 gap	4	10	17760270
23	ISB11 gap	4	10	17760300
24	DMV11 gap	8	20	17760320
25	DEUNA gap	4	10 *	17760330
26	UDA50/RQDX1 gap	2	4 *	17760334
27	DMF32 gap	16	40	17760340
28	KMS11 gap	6	20	17760360
29	VS100 gap	8	20	17760400
30	TU81 gap	2	4	17760404
31	KMV11 gap	8	20	17760420
32	DHV11/DHU11/ CXA16/CXB16 gap	8	20	17760440

* The first device of this type has a fixed address. Any extra devices have a floating address.

** The first two devices of this type have a fixed address. Any extra devices have a floating address.

The address assignment rules are as follows.

1. Addresses, starting at 17760010(octal) for Q-bus systems, are assigned according to the sequence of Table B-1.

2. Option and gap addresses are assigned according to the octal modulus as follows.

   - Devices with an octal modulus of 4 are assigned an address on a 4(octal) boundary (the two lowest-order address bits = 0)

   - Devices with an octal modulus of 10 are assigned an address on a 10(octal) boundary (the three lowest-order address bits = 0)

   - Devices with an octal modulus of 20 are assigned an address on a 20(octal) boundary (the four lowest-order address bits = 0)

   - Devices with an octal modulus of 40 are assigned an address on a 40(octal) boundary (the five lowest-order address bits = 0)

3. Address space equal to the device's modulus must be allowed for each device which is connected to the bus.

4. A 1-word gap, assigned according to rule 2, must be allowed after the last device of each type. This gap could be bigger when rule 2 is applied to the following rank.

5. A 1-word gap, assigned according to rule 2, must be allowed for each unused rank on the list if a device with a higher address is used. This gap could be bigger when rule 2 is applied to the following rank.

If extra devices are added to a system, the floating addresses may have to be reassigned in agreement with these rules.

B.2 FLOATING VECTORS

Each device needs two 16-bit locations for each vector. For example, a device with one receive and one transmit vector needs four words of vector space.

The vector assignment rules are as follows:

1. Each device occupies vector address space equal to 'Size' words. For example, the DLV11-J occupies 16 words of vector space. If its vector were 300(octal), the next available vector would be at 340(octal).

2. There are no gaps, except those needed to align an octal modulus.

Table B-2 Floating Vector Address Assignments

Rank	Device	Size (Decimal)	Modulus (Octal)	
1	DC11	4	10	
1	TU58	4	10	
2	KL11	4	10	*
2	DL11-A	4	10	*
2	DL11-B	4	10	*
2	DLV11-J	16	10	
2	DLV11, DLV11-F	4	10	
3	DP11	4	10	
4	DM11-A	4	10	
5	DN11	2	4	
6	DM11-BB/BA	2	4	
7	DH11 modem control	2	4	
8	DR11-A, DRV11-B	4	10	
9	DR11-C, DRV11	4	10	
10	PA611 (reader + punch)	8	10	
11	LPD11	4	10	
12	DT07	4	10	
13	DX11	4	10	
14	DL11-C to DLV11-E	4	10	
15	DJ11	4	10	

* If a KL11 or DL11 is used as the console, it has a fixed vector.

Table B-2 Floating Vector Address Assignments (Cont.)

Rank	Device	Size (Decimal)	Modulus (Octal)	
16	DH11	4	10	
17	VT40	8	10	
17	VSV11	8	10	
18	LPS11	12	10	
19	DQ11	4	10	
20	KW11-W, KWV11	4	10	
21	DU11, DUV11	4	10	
22	DUP11	4	10	
23	DV11 + modem control	6	10	
24	LK11-A	4	10	
25	DWUN	4	10	
26	DMC11/DMR11	4	10	
27	DZ11/DZS11/DZV11, DZ32	4	10	
28	KMC11	4	10	
29	LPP11	4	10	
30	VMV21	4	10	
31	VMV31	4	10	
32	VTV01	4	10	
33	DWR70	4	10	
34	RL11/RLV11	2	4	*
35	TS11, TU80	2	4	*
36	LPA11-K	4	10	
37	IP11/IP300	2	4	*
38	KW11-C	4	10	
39	RX11/RX211 RXV11/RXV21	2	4	*
40	DR11-W	2	4	
41	DR11-B	2	4	*
42	DMP11	4	10	
43	DPV11	4	10	
44	ML11	2	4	**

* The first device of this type has a fixed vector. Any extra devices have a floating vector.

** ML11 is a MASSBUS device which can connect to UNIBUS via a bus adapter.

Table B-2 Floating Vector Address Assignments (Cont.)

Rank	Device	Size (Decimal)	Modulus (Octal)	
45	ISB11	4	10	
46	DMV11	4	10	
47	DEUNA	2	4	*
48	UDA50/RQDX1	2	4	*
49	DMF32	16	4	
50	KMS11	6	10	
51	PCL11-B	4	10	
52	VS100	2	4	
53	TU81	2	4	
54	KMV11	4	10	
55	KCT32	4	10	
56	IEX	4	10	
57	DHV11/DHU11/CXA16/CXB16	4	10	

* The first device of this type has a fixed vector. Any extra devices have a floating vector.


APPENDIX C

APPENDIX C
AUTOMATIC FLOW CONTROL

C.1 OVERVIEW

Flow control is the control of the flow of data along a communications line, to prevent an overspill of queues or buffers, or to prevent the loss of data which the receiver is unable to accept.

The method of flow control adopted for the CXA16/CXB16 is datastream-embedded ASCII control characters. The control characters used are XOFF (octal 023) and XON (octal 021). XOFF stops transmission and XON starts transmission. The codes are transmitted in the opposite direction to that of the data they control.

The CXA16/CXB16 has one mode of operation for transmitted data (received flow-control characters) and two modes of operation for received data (transmitted flow-control characters). Each mode can be enabled on a 'per channel' basis. Each direction of flow is discussed separately within this appendix.

C.2 CONTROL OF TRANSMITTED DATA

The transmitted-data mode of flow control is the simplest of the three flow-control modes of the CXA16/CXB16.

When the CXA16/CXB16 receives an XOFF character for a particular channel, the TX.ENA bit for that channel is cleared. When this bit is clear the CXA16/CXB16 will not transmit any data on that channel; however, internally generated flow-control characters will still be transmitted. When an XON character is received, the TX.ENA bit for that channel is set. Figure C-1 illustrates the operation of the transmitted data flow control.

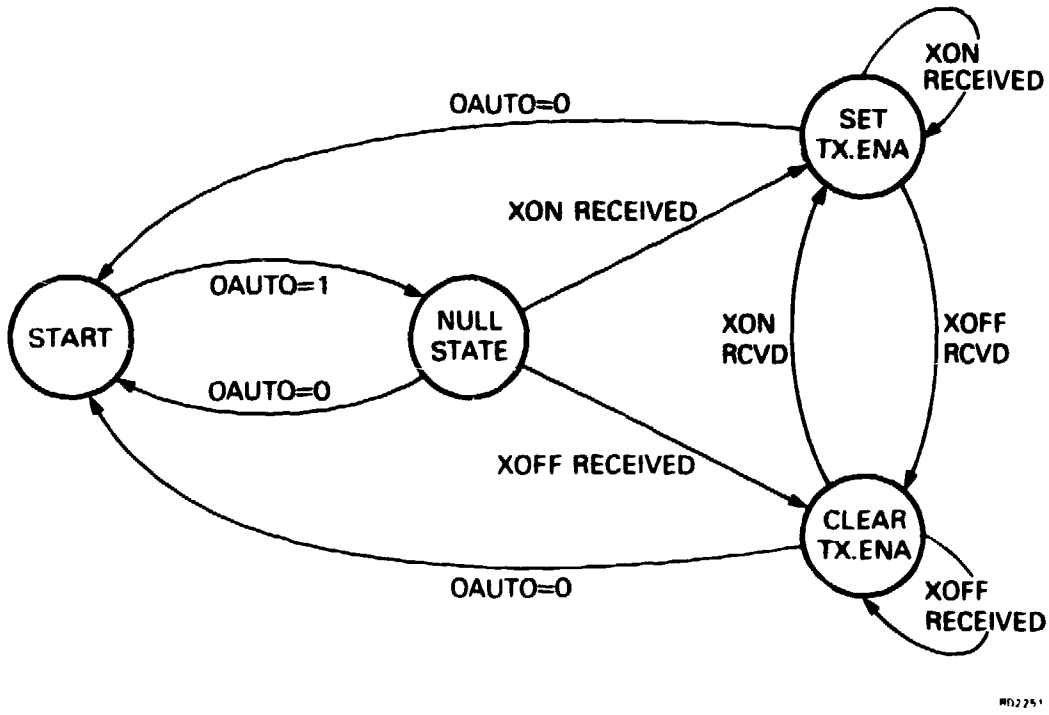


Figure C-1 Transmitted Data Flow Control

Only characters without transmission errors are checked for XON and XOFF codes. The characters have their parity bit stripped before comparison.

NOTE

For the automatic flow control to operate correctly, the terminal must also recognize and respond to flow control characters.

The transmitted-data mode of flow control is enabled by setting OAUTO (bit 4 of the line-control register), and is disabled by clearing it. The default for this mode is disabled. The CXA16/CXB16 can alter the state of the TX.ENA bit, up to 50 microseconds after the program clears the OAUTO bit.

Received flow-control characters are processed in the same way as normal characters, and are placed into the receive FIFO. This is not affected by OAUTO; but these characters can be filtered out by setting DISAB.XRPT. If DISAB.XRPT is set, you do not need a routine in your software driver to filter flow-control characters from the data stream.

C.3 CONTROL OF RECEIVED DATA

Received data flow control is slightly more complicated than transmitted data flow control. Therefore the two modes of received data flow control are described separately.

C.3.1 Flow Control by the Level of the Receive FIFO

Occasionally, the program may not be able to empty the receive FIFO as fast as the received data is filling it. Because the program does not know how full the receive FIFO is, it cannot take action to prevent data loss. To overcome this problem, the CXA16/CXB16 can be programmed on a 'per channel' basis. When the receive FIFO becomes three-quarters full, an XOFF is sent to the channels from which data is received. An XOFF character is then sent in response to every second received character, until the receive FIFO level drops below half full. An XON character is then transmitted. The operation of receive FIFO-level flow control is shown in Figure C-2.

The receive FIFO-level flow-control mode is enabled by setting IAUTO (bit 1 of the line-control register), and disabled by clearing the bit. The default for this mode is disabled. If IAUTO is cleared after an XOFF is sent, but before the receive FIFO level drops below half full, an XON is still sent.

NOTE

FIFO.CRIT is set (T) when the receive FIFO contains 192 characters and is cleared (F) when the receive FIFO contains less than 128 characters.

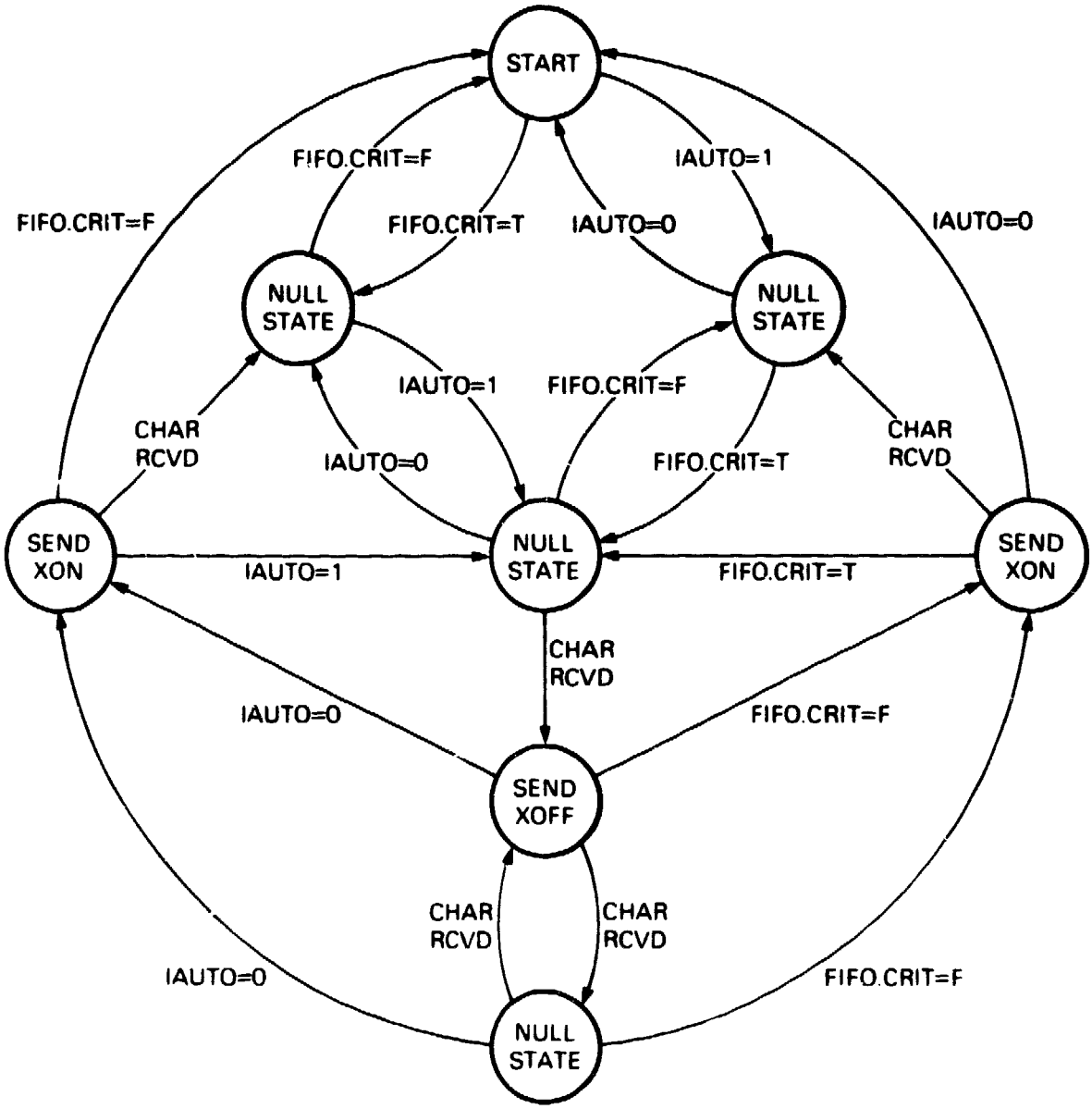


Figure C-2 Receive FIFO-Level Flow Control

C.3.2 Flow Control by Program Initiation

Occasionally, the program itself may need to invoke flow control automatically, for example, when internal buffers become full. To allow this, the CXA16/CXB16 has a FORCE.XOFF bit (bit 5 of the line-control register). When the FORCE.XOFF bit is set, the CXA16/CXB16 transmits an XOFF character for that channel. A further XOFF bit is transmitted for every second character received on the channel afterwards. An XON is sent when the FORCE.XOFF bit is cleared. Figure C-3 shows the operation of program-initiated flow control.

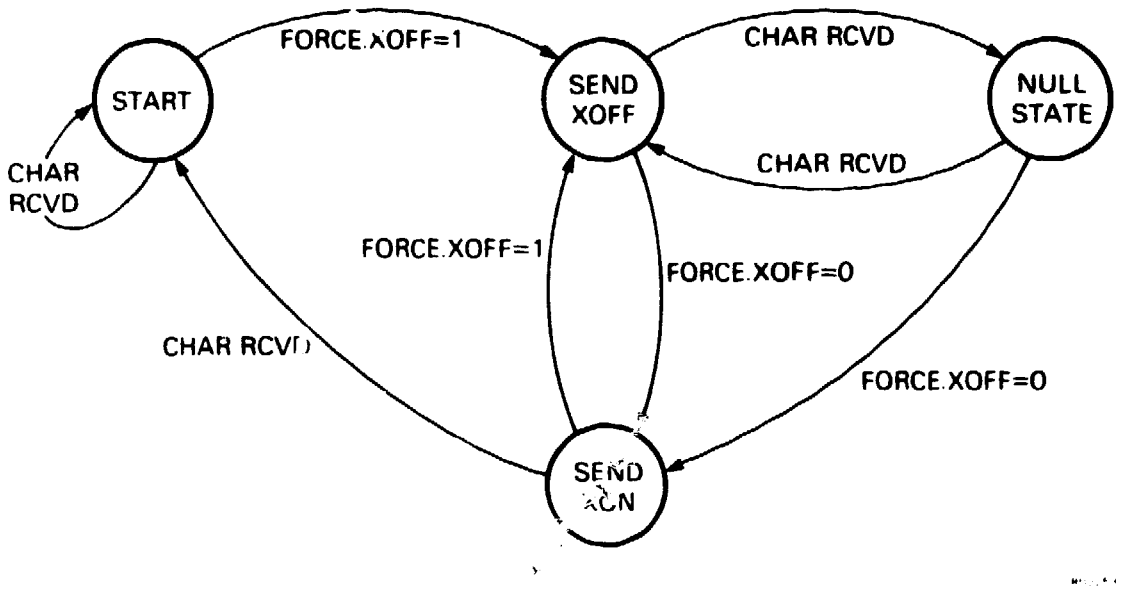


Figure C-3 Program-Initiated Flow Control

NOTE

If the program sets the FORCE.XOFF bit and then immediately clears it, the XOFF code will not be transmitted immediately. This is because there is a delay before the sequencer sees the program request and acts on it.

The FORCE.XOFF bit is cleared by a CXA16/CXB16 reset sequence.

C.3.3 Mixing the two Types of Received Data Flow Control

To calculate the effect of using the two modes, they should be logically ORed together; an XON will not be sent until both sources are inactive. An XOFF will be sent when FORCE.XOFF is set, even if FIFO-critical mode is active and an XOFF has already been sent on that channel. If the receive FIFO critical mode becomes active whilst FORCE.XOFF is set, then another XOFF is sent in response to the next received character.


APPENDIX D

APPENDIX D
GLOSSARY OF TERMS

D.1 SCOPE

This appendix contains a glossary of terms used in this manual and in other DIGITAL technical manuals in this series. The terms are in alphabetical order for easy reference.

D.2 GLOSSARY

Asynchronous. A method of serial 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-flow. Automatic flow control. A method by which the CXA16/CXB16 controls the flow of data by means of special characters within the data stream.

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

Base address. The Q-bus address of the first (lowest) device register (CSR).

BMP. Background Monitor Program.

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

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

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

EIA. Electrical Industries Association. An American organization with the same function as the **CCITT.**

FCC. Federal Communications Commission. An 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. An address assigned to an option which does not have a fixed address allocated. The address is dependent on other floating address devices connected to the bus.

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

FRU. Field-Replaceable Unit.

IC. Integrated Circuit.

I/O. Input/Output.

LSB. Least-Significant Bit.

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

MMJ. Modified Modular Jack.

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.**

MSB. Most Significant Bit.

Multiplexer. A device which allows a number of inputs to share one common output.

Null modem. A cable which allows two terminals which use **modem** control signals to be connected toge ier directly. It is only possible over short distances.

OCTART. A single IC containing eight **UARTs.**

PCB. Printed Circuit Board.

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

PSTN. Public Switched Telephone Network.

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

RAM. Random Access Memory.

RFI. Radio Frequency Interference.

ROM. Read-Only Memory.

Split-speed. A facility of a data communications channel which can transmit data at a different speed from the received data.

UART. Universal Asynchronous Receiver Transmitter. A device which converts between serial and parallel data, used for transmission and reception of serial **asynchronous** data on a channel.

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

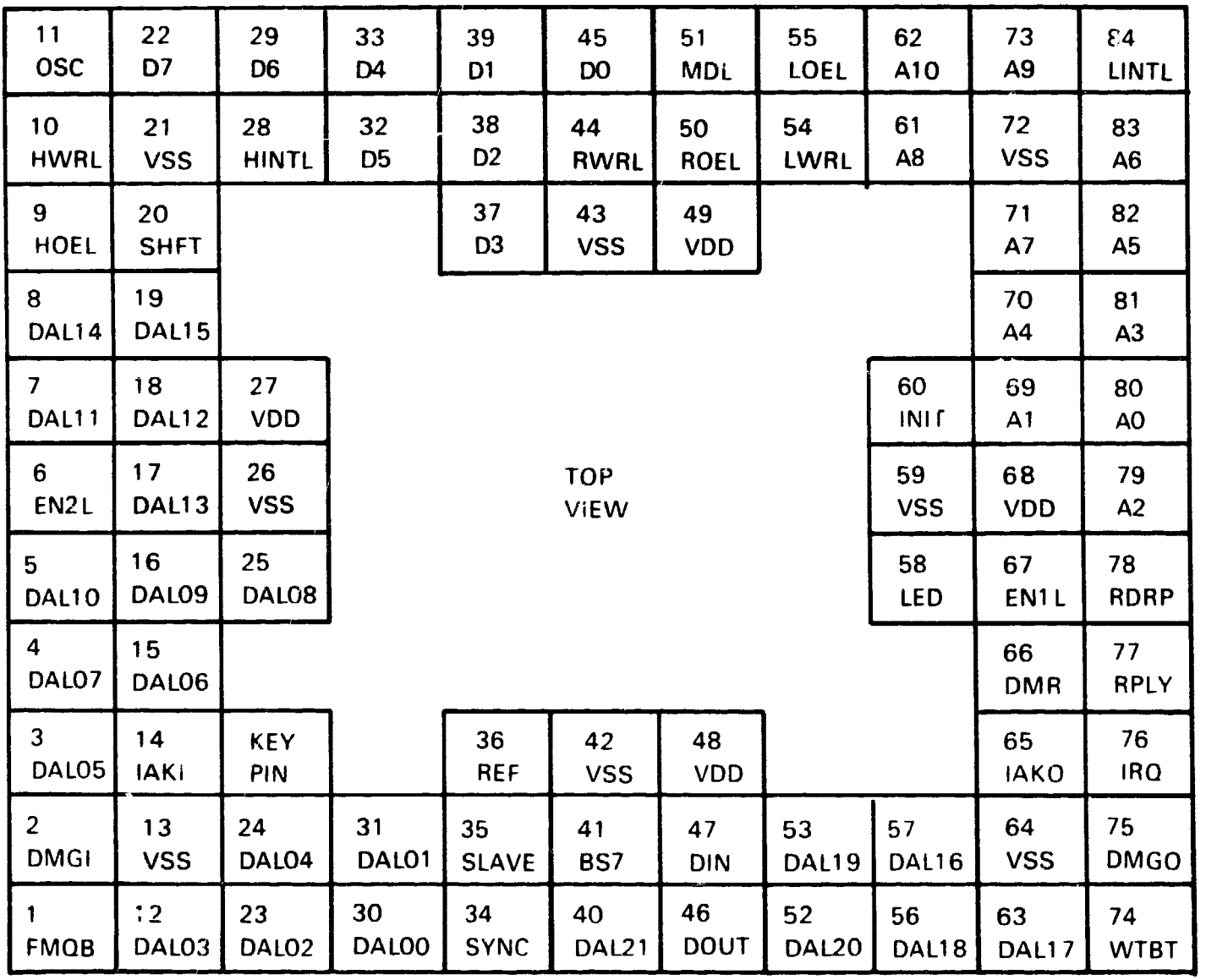
XON. A control code (21(octal)) used to enable a transmitter which has been disabled by an **XOFF** code.


APPENDIX E

APPENDIX E
CONTROL CHIP AND OCTART

E.1 SCOPE

This appendix describes the signals associated with the control chip and the OCTART. It is to be used in conjunction with the maintenance printset.



MKV88-2057

E.2 CONTROL CHIP

Table E-1 lists only the interface signals; it does not include internal signals.

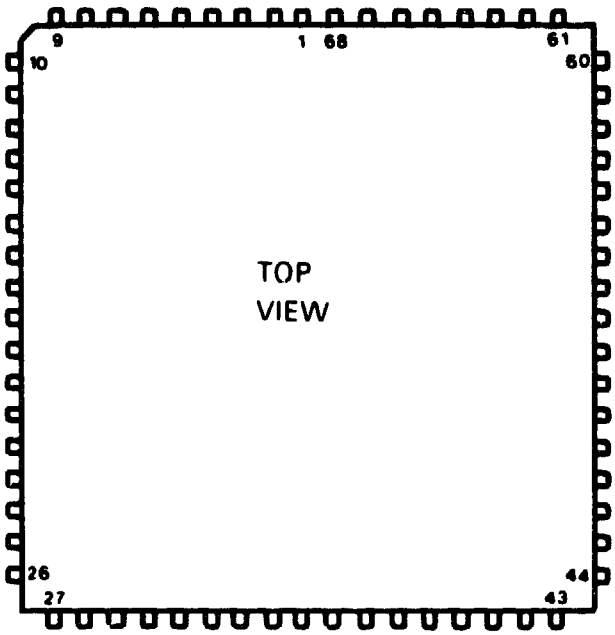
Table E-1 Control-Chip Interface Signals

Signal Name	Description
SLAVE.H	Controls the direction of Q-bus transceiver E9. Deasserted during bus mastership since TSACK.H is a pull-up, this asserts BSACK1 on the Q-bus
RPLY.H DMR.H IRQ.H DMGO.H IAKO.H SYNC.H BS7.H DIN.H REF.H DMGI.H IAKI.H WTBT.H DOUT.H INIT.H	) Q-bus signals
FMQB.H	From Q-bus (normally high) -- affects the direction of DA.L<15:0>
RDRP.H	Normally high, it is set low to assert RDRP and DMA address lines
EN1.L	Normally low, is set high to disable the Q-bus transceiver when RDRP.H is changing state
EN2.L	Normally low, is set high to disable the Q-bus transceiver when SLAVE.H is changing state
DAL<21:0>.H	Buffered Q-bus data and address lines
LED.H	Self-test output to LED (through DC021 driver)
SHFT.H	Shift register input (from switchpack shift registers)

Table E-1 Control-Chip Interface Signals (Cont.)

Signal name	Description
A<10:0>.H	RAM/OCTART address bus
D<7:0>.H	RAM/OCTART data bus
ROE.L	RAM output enable - active low (read strobe)
RWR.L	RAM write strobe - active low
HOE.L	High-OCTART read strobe
HWR.L	High-OCTART write strobe
LOE.L	Low-OCTART read strobe
LWR.L	Low-OCTART write strobe
HINIT.L	High-OCTART interrupt to control chip
LINIT.L	Low-OCTART interrupt to control chip
MDL	Modem latch write (not used on CXA16/CXB16)
OSC.H	14.7456 MHz oscillator input

E.3 OCTART



MKV88-2058

Table E-2 OCTART Signals

Signal Name	Description
D<7:0>.H	RAM/OCTART data bus
A<1:0>.H	Read-only address bus (register select)
TEST.L	Used for manufacturing test
MDM.L	Modem support - held disabled on CXA16/CXB16
OE.L	Read strobe
WR.L	Write strobe
OSC.H	14.7456 MHz oscillator input
INT.L	Interrupt output
SDO<7:0>.H	Serial data out
SDI<7:0>.H	Serial data in
CTS<7:0>.L	Clear to send
DSR<7:0>.L	Data set ready
DCD<7:0>.L	Data carrier detect
RI<7:0>.L	Ring indicator


INDEX

INDEX

Active adapter, H3105, 1-11
Address, device, 2-2, 2-3, B-1, B-2, B-3
Assignment, RAM, 5-3

Background monitor program, 3-16, 3-39, 4-4, 4-6
Baud rates, 1-2
BC16D-25 interconnecting cable, 1-3, 2-1, 2-9
Break, 3-23
Buffers, Q-bus, 1-11
Bus grant continuity, 2-6

Cable, BC16D-25 interconnecting, 1-3, 2-1, 2-9
    H3104, concentrator, 1-3, 1-11, 2-1, 2-9
    lengths, 1-10, 1-11
Characteristics, electrical, 1-2
Concentrator, H3104 cable, 1-3, 2-1, 2-9
Conditions, environmental, 1-8
Configurations, 1-5, 1-6
Connections, Q-bus, A-1
Connector, H8521 passive, 1-11
Connectors, 1-3, 1-11, 1-13, 4-2
    loopback test, 1-3, 2-1, 2-12, 2-14
    MMJ, 1-11
Continuity, bus grant, 2-6
Control chip, 1-4, 1-13, 1-14, 1-15, 3-37, 5-1, 5-7, 5-12, 5-15, E-2
    flow, C-1, C-2

Data rates, 1-9
DEC422, 1-1, 1-10, 1-12
DEC423, 1-1, 1-10, 1-11, 1-12
Device address, 2-2, 2-3, B-1, B-2, B-3
DHU11 mode, 1-1, 1-2, 1-5, 1-15, 2-2, 2-3, 2-4, 3-19, 3-20, 5-16
DHV11 mode, 1-1, 1-2, 1-5, 1-15, 2-2, 2-3, 2-4, 3-19, 3-20, 5-16
Diagnostic programs, 1-3
Diagnostics, 2-7, 2-9, 2-10
DMA, 1-1, 1-15, 2-7, 3-6, 3-24, 3-25, 3-26, 3-27, 3-31, 3-44, 5-15
Drivers, Q-bus, 5-1, 5-13

EIA-232-D, 1-9, 1-10, 1-11
Electrical characteristics, 1-2
    requirements, 1-8
Environmental conditions, 1-8
EOS/ESD, protection, 1-10, 1-12

FIFO, 64-character transmit, 1-1
    receive, 1-2, 1-3, 1-15
Flow control, 1-2, C-1, C-2

Grant, bus, continuity, 2-6

H3104 cable concentrator, 1-3, 1-11, 2-1, 2-9
H3105 active adapter, 1-11
H8521 passive connector, 1-11

Installation, 2-2, 2-8
Interconnecting, BC16D-25, cable, 1-5, 2-1, 2-9
Interface, line, 5-1, 5-13
    Q-bus, 5-7, 5-15
    serial line, 1-2, 1-13
    standards, 1-9
Interrupt request, 2-7, 3-33
    vector, 2-5, B-5, B-6, B-7, B-8

Lengths, cable, 1-10, 1-11
Line interface, 5-1, 5-13
    receivers, 1-12, 1-13
    serial, interface, 1-2, 1-13
    transmitters, 1-12, 1-13
Loads, Q-bus, 1-8
Loopback test connectors, 1-3, 2-1, 2-14, 2-12, 4-2, 4-3, 4-4, 4-5, 4-9, 4-10, 4-11, 4-12

Maintenance kit, 1-3
MicroVAX II, 1-3, 2-11, 2-12, 4-5, 4-11
MMJ connectors, 1-11
Monitor, background, program, 3-16, 3-39, 4-4, 4-6

OCTART, 1-12, 1-15, 1-16, 3-23, 3-37, 5-1, 5-5, 5-8, 5-11, E-1, E-4

Parity, 3-15
Passive, H8521, connector, 1-11
PDP-11, 1-3, 2-11, 4-5
Program, background monitor, 3-16, 3-39, 4-4, 4-6
Programs, diagnostic, 1-3
Protection, EOS/ESD, 1-10, 1-12

Q-bus, 1-1, 1-3, 1-5, 2-6, 5-8, 5-10
    buffers, 1-13
    connections, A-1
    drivers, 5-1, 5-13
    interface, 1-15, 5-7
    loads, 1-8
    receivers, 5-1, 5-13

RAM assignment, 5-4
Rates, baud, 1-2
    data, 1-9
Receive FIFO,1-2, 1-3, 1-15
Receivers, line, 1-12, 1-13
    Q-bus, 5-1, 5-11
Registers, 3-1
Requirements, electrical, 1-8
RS-422-A, 1-1, 1-2, 1-9, 1-12
RS-423-A, 1-1, 1-2, 1-9, 1-10

Self-test, 1-2, 1-3, 1-5, 1-15, 2-9, 3-6, 3-8, 3-29, 3-37, 3-38
           4-3, 5-8
Serial-line interface, 1-2, 1-13
Shift register, 5-10
Standards, interface, 1-9
Static, 1-2
Switchpacks, 1-5, 1-13, 2-2, 2-3, 2-4, 2-5, 5-9

Throughput, 1-2, 1-3, 1-9
Transmit, 64-character, FIFO, 1-1
Transmitters, line, 1-12, 1-13

User-mode, 4-4

Vector, interrupt, 2-5, B-5, B-6, B-7, B-8

Xoff, 1-2, 1-3, 3-13, 3-22, 3-23, 3-24, 3-34, C-1, C-2
Xon, 1-2, 1-3, 3-13, 3-22, 3-23, 3-24, 3-34, C-1, C-2