EK-KXJCA-UG-001

KXJ11-CA Single-Board Computer

User's Guide

digital™

EK-KXJCA-UG-001

KXJ11-CA Single-Board Computer

User's Guide

Prepared by Educational Services
of Digital Equipment Corporation

Preliminary Edition, May 1986
First Edition, January 1987

The information in this document is subject to change without notice. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts 01754.

digital™
DEC
DECmate
DECUS
DECwriter
DIBOL
MASSBUS
PDP
P/OS
Professional
Q-Bus
Q22-Bus
Rainbow
RSTS
RSX
RT
UNIBUS
VAX
VAXstation
VAXstation II
VMS
VT
Work Processor

CONTENTS

CHAPTER 1 OVERVIEW

CHAPTER 2 INSTALLATION

CHAPTER 3 ARCHITECTURE

CHAPTER 4 DMA TRANSFER CONTROLLER

CHAPTER 5 PARALLEL I/O CONTROLLER

CHAPTER 6 SERIAL LINE UNITS

APPENDIX A MEMORY MAP SUMMARY

APPENDIX B KXJ11-CA/KXT11-CA DIFFERENCES

APPENDIX C USER (P)ROM PROGRAMMING INFORMATION

APPENDIX D BUILDING AN APPLICATION

FIGURES

Figure No. Title Page No.

TABLES

CHAPTER 1
OVERVIEW

1.1 INTRODUCTION

The KXJ11-CA (M7616) is an I/O processor based on the J-11 microprocessor chip. It is a quad-height, extended length, single-width module that executes the extended PDP-11 instruction set (all 140 instructions including floating-point) with memory management. The KXJ11-CA can operate as a Q-Bus slave device under the direction of a Q-Bus arbiter processor, or can act as a standalone processor.

The KXJ11-CA meets the specification for a Q-Bus slave and Q-Bus DMA master, and can interface with most of Digital's large family of Q-Bus modules described in the Microcomputer Interfaces Handbook and the Microcomputers and Memories Handbook.

1.2 KXJ11-CA HARDWARE FEATURES

The KXJ11-CA has the following features:

- J-11 (DCJ11-AC) 16-bit microprocessor

  Executes extended PDP-11 instruction set (140 instructions including floating-point)

  Contains memory management unit for three levels of memory protection and 4 MB addressing

  Operates at 14 MHz

- Memory

  512 KB of dynamic RAM

  Can be accessed by local (on-board) devices and Q-Bus devices

  Up to 64 KB of PROM, 16 KB of which is allocated for firmware

- Q-Bus interface

    16 word, two-ported RAM (TPR) register file for passing commands and parameters

    Mechanism for posting interrupts to the Q-Bus

- Two channel programmable DMA transfer controller (DTC)

    Performs transfers between local 22-bit addresses and 16-bit, 18-bit, or 22-bit Q-Bus addresses

- Eight control/status registers

- Console asynchronous serial line

    DL-compatible

    EIA RS-422/RS-423/RS-232C-compatible

    Programmable baud rates of 300 to 38400 baud

- Primary synchronous/asynchronous serial line unit

    Full modem support

    EIA RS-449 (CCITT V.24) and RS-422/RS-423/RS-232C-compatible

    Programmable baud rates of 110 to 76800 baud

    Bit-oriented or character-oriented synchronous protocol support

- Secondary synchronous/asynchronous serial line unit

    RS-449 (CCITT V.24) data and timing only

    RS-422/RS-423/RS-232C-compatible

    Programmable baud rates of 110 to 76800 baud

    Bit-oriented or character-oriented synchronous protocol support

    Party line operation

- Two programmable timers for the synchronous/asynchronous serial line units and one watchdog timer

- Parallel I/O Interface

  Two 8-bit bidirectional double-buffered I/O ports

  One 4-bit special purpose I/O port

  Pattern recognition logic

  Three independent 16-bit counter/timers

  IEEE 488 electrically-compatible

1.3 KXJ11-CA OPERATING MODES

The KXJ11-CA can operate in either standalone mode or in peripheral processor mode. The sections that follow describe these modes. The AC and DC characteristics of the KXJ11-CA are identical in both modes.

1.3.1 Standalone Mode

The KXJ11-CA can be configured to operate as a standalone processor. In standalone mode, communication with other Q-Bus devices (including the system arbiter) is disabled. The backplane into which the KXJ11-CA is plugged acts as a source of both power and ground. The KXJ11-CA preserves the continuity of the daisy-chained interrupt acknowledge and DMA grant lines on the backplane.

To select standalone mode, position the on-board ID switch at 0 or 1. In the operational descriptions that appear in this and other chapters, ignore any references to Q-Bus activity if the KXJ11-CA is operating in standalone mode.

1.3.2 Peripheral Processor Mode

The KXJ11-CA is designed primarily as an I/O processor. In a typical system, a KXJ11-CA is connected to one or more I/O devices that would otherwise be interfaced directly with the Q-Bus. In peripheral processor mode, the KXJ11-CA handles interrupts and data processing associated with the I/O devices, freeing the Q-Bus from traffic that would ordinarily degrade system performance. Up to 14 peripheral processors per system can be accommodated. Peripheral processor mode is selected when the on-board ID switch is in positions 2 through 15.

1.4 KXJ11-CA SPECIFICATIONS

Physical	
Height (quad)	26.6 cm (10.5 in)
Length (extended)	22.8 cm (8.9 in) (includes module handle)
Width (single)	1.27 cm (0.5 in)
Weight	665 g (22 oz) maximum
Power requirements	
Operational power	+5V ± 5% 6.0 A maximum +12V ± 5% 1.4 A maximum with DLV11-KA option +12V ± 5% 0.4 A maximum without DLV11-KA option
Bus loads	AC loads = 2 units DC loads = 1 unit
Environmental	
Temperature	
Storage	-40 to 66°C (-40 to 150°F)
Operating	5 to 60°C (41 to 140°F)
Relative humidity	
Storage	10% to 90% (non-condensing)
Operating	10% to 90% (non-condensing)
Altitude	
Storage	Up to 15 km (50,000 ft)
Operating	Up to 15 km (50,000 ft)
Air quality	Air must be non-caustic.

1.5 TERMINOLOGY USED IN THIS DOCUMENT

Some terms used throughout this document are defined below.

Local device/memory -- Refers to an I/O device or memory that is located on the KXJ11-CA board.

Global or Q-Bus -- Refers to any Q-Bus address, including KXJ11-CA Q-Bus addresses.

Shared memory -- Refers to the area of memory in local address space that is also assigned to a Q-Bus address range.

Arbiter -- The Q-Bus default master, interrupt acknowledger, DMA grantor, and power up/down and reset control device (usually resides in the first slot of the Q-Bus).

BDAL bus -- The Q-Bus (backplane) multiplexed data and address lines.

Q-Bus transceivers -- The interface between the Q-Bus and the QDAL bus.

ZDAL bus -- The 22-bit address and 16-bit data path between the BDAL transceivers and the KDAL transceivers.

KDAL bus -- The KXJ11-CA internal module multiplexed data and address bus, which is common to all local memory and I/O devices.

JDAL bus -- The 22-bit address and 16-bit data path between the KDAL transceivers and the J-11 microprocessor.

Instruction cycle -- The sequence of bus transactions involved in the execution of an entire instruction by the J-11 microprocessor.

Transaction -- Either a KXJ11-CA address and data exchange or a DMA master address and data exchange with the necessary handshake signal assertions.

DTC -- Refers to the Z8016 direct memory access transfer controller.

PIO -- Refers to the Z8036 parallel I/O unit and counter/timer.

uPD7201 -- Refers to the NEC 7201 multiprotocol serial controller. Also referred to as SLU2.

Native firmware -- ROM-based program that directs and coordinates the operation of the KXJ11-CA, and allows the KXJ11-CA to interpret and respond to commands from the arbiter processor.

DLART -- Refers to the DL-compatible asynchronous receiver/transmitter used as the console port. Also referred to as SLU1.

1.6 RELATED SOFTWARE PRODUCTS

The toolkits listed below support the development of applications on the KXJ11-CA. Refer to the Software Product Descriptions (SPDs) for more detailed information.

Peripheral Processor Tool Kit - RT-11, Version 2.3	SPD 12.70
Peripheral Processor Tool Kit - RSX, Version 2.3	SPD 13.25
MicroPower/Pascal-RSX, Version 2.3 Modular Executive and Microcomputer Software Development Toolset	SPD 14.83
MicroPower/Pascal-Micro/RSX, Version 2.3 Modular Executive and Microcomputer Software Development Toolset	SPD 18.21

Peripheral Processor Tool Kit - Micro/RSX, Version 2.3	SPD 18.48
MicroPower/Pascal-RT, Version 2.3 Modular Executive and Microcomputer Software Development Toolset	SPD 19.12
MicroPower/Pascal-VMS, Version 2.3 Modular Executive and Microcomputer Software Development Toolset	SPD 26.24

1.7 AVAILABLE OPTIONS

The following option is available with the KXJ11-CA.

DLV11-KA -- EIA to 20 mA converter. This option consists of the DLV11-KB EIA-to-20 mA converter unit and a BC21A-03 EIA interface cable. Detailed information is provided in the DLV11-KA EIA to 20 mA Installation Guide (EK-DLVKA-IN), the DLV11-KA Maintenance Print Set (MP00694), and the Microcomputer Interfaces Handbook (EB-23144-18).

1.8 RELATED DOCUMENTS

This User's Guide is the primary reference in the documentation package that accompanies the KXJ11-CA. The other documents in the package include:

DCJ11 Microprocessor User's Guide
AmZ8036/AmZ8536 Counter/Timer, Parallel I/O Technical Manual
uPD7201 Multiprotocol Serial Controller Technical Manual
Am9516/AmZ8016 DMA Controller Technical Manual
8254 Programmable Interval Timer Data Sheet
DLART Data Sheet
KXJ11-CA Schematics
KXJ11-CA Firmware Listings

The order number for the documentation package is EK-KXJCA-DK, if additional sets are required. The documentation package is also referred to as the KXJ11-CA Hardware Documentation Kit.

Other documents the reader may find useful include:

Title	**Order Number**
Microcomputers and Memories Handbook	EB-20912-20
Microcomputer Interfaces Handbook	EB-23144-18
PDP-11 Architecture Handbook	EB-23657-18
TU58 Technical Manual	EK-OTU58-TM
DLV11-KA EIA to 20 mA Installation Guide	EK-DLVKA-IN
DLV11-KA Maintenance Print Set	MP00694

These documents are available from:

Digital Equipment Corporation
Accessories and Supplies Group
P.O. Box CS2008
Nashua, NH 03061

Attention: Documentation Products

CHAPTER 2
INSTALLATION

2.1 INTRODUCTION

This chapter describes how to install the KXJ11-CA module.

> **NOTE**
>
> Before changing the factory-shipped jumper configuration, make sure the jumpers match the jumpers shown in Figure 2-1, and verify that the module is operating as described in Section 2.7.

Installation includes the following activities.

1. Selecting operating characteristics and installing appropriate jumpers.

2. Determining power supply requirements.

3. Installing the board into a backplane.

4. Selecting and connecting cables from serial and parallel I/O interfaces to external devices.

5. Verifying proper operation.

2.2 SELECTING OPERATING FEATURES

Several characteristics of the KXJ11-CA are defined by jumper settings. This section describes the characteristics that are part of the factory-shipped configuration. It also shows how to change these characteristics by changing the appropriate jumpers.

Figure 2-1 illustrates the factory-shipped jumper settings. Table 2-1 summarizes the meaning of each jumper setting. The sections that follow describe the various jumper setting alternatives available.

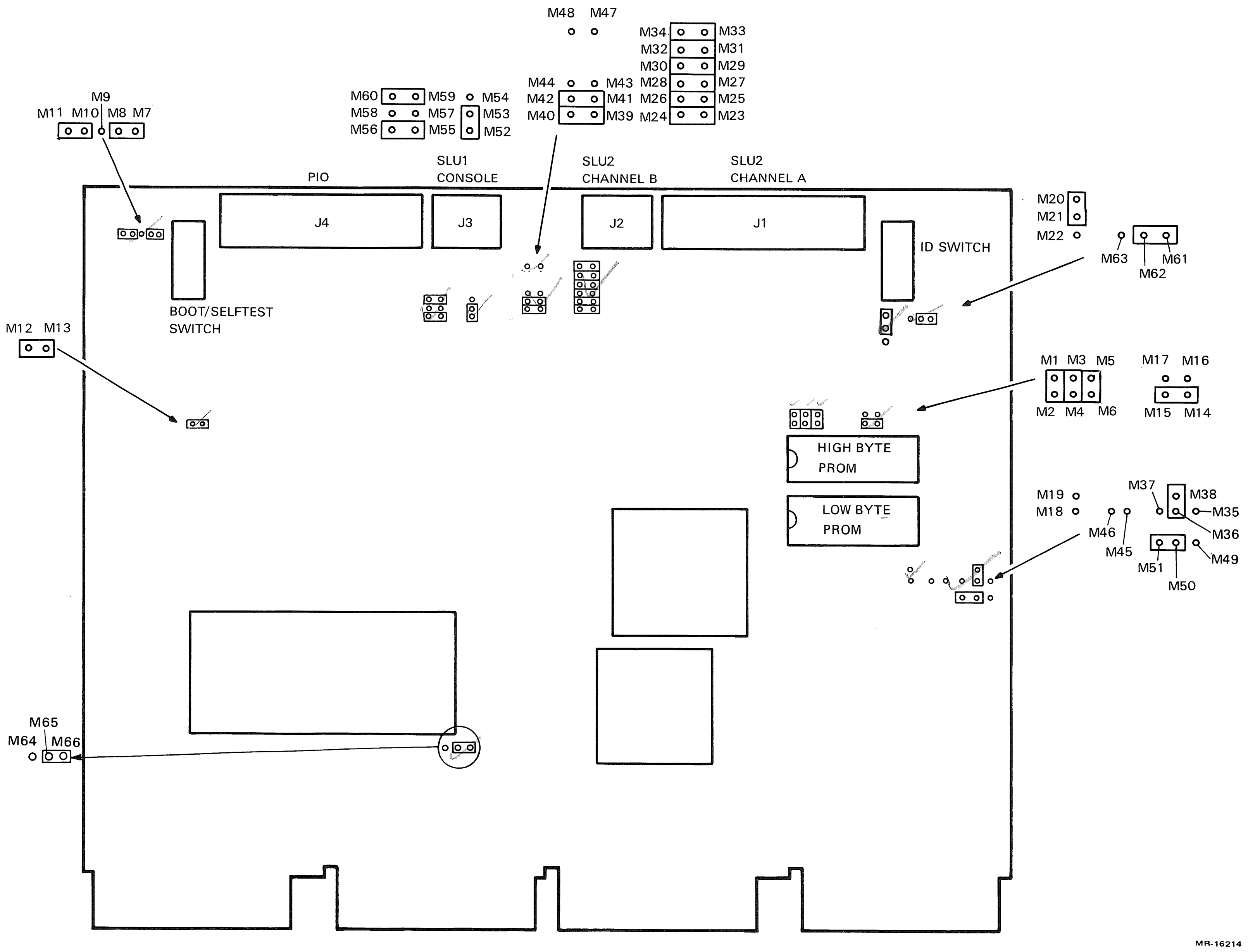


Figure 2-1 KXJ11-CA Jumper Layout

Table 2-1 Factory-Shipped Jumper Configuration

Function	Setting	Jumpers Installed
Q-Bus Size	22 bits	M3 to M4 M5 to M6
Q-Bus Base Address	17775400	M1 to M2
ID Switch Position	8	
DMA Requests		
SLU2 Channel A Receiver	Enabled	M10 to M11
8036 Counter/Timer	Disabled	
SLU2 Channel A Transmitter	Enabled	M7 to M8
Locked Instruction Enable	Disabled	M65 to M66
BREAK Enable	Enabled	M12 to M13
HALT Option Selection	MicroODT	M14 to M15
Power-Up Option Selection	Firmware	No jumper
PROM Addressing	15-bit	No jumper
SLU1 Baud Rate	9600	M56 to M55 M60 to M59
SLU1 Transmitter	RS423	M62 to M61
SLU1 Receiver	RS423	No jumper
SLU2 Channel A Receiver	RS422	M34 to M33 M32 to M31 M30 to M29 M28 to M27 M26 to M25 M24 to M23
SLU2 Channel B Transmitter	RS422	M38 to M36 M51 to M50
SLU2 Channel B Receiver	RS422	M42 to M41 M40 to M39 M20 to M21
Real-Time Clock Interrupt	60 Hz	M52 to M53
Boot/Self-test Switch Position	6	

Note: The SLU2 Channel A Transmitter is not configured with jumpers, but is configured by selecting appropriate signals on connector J1.

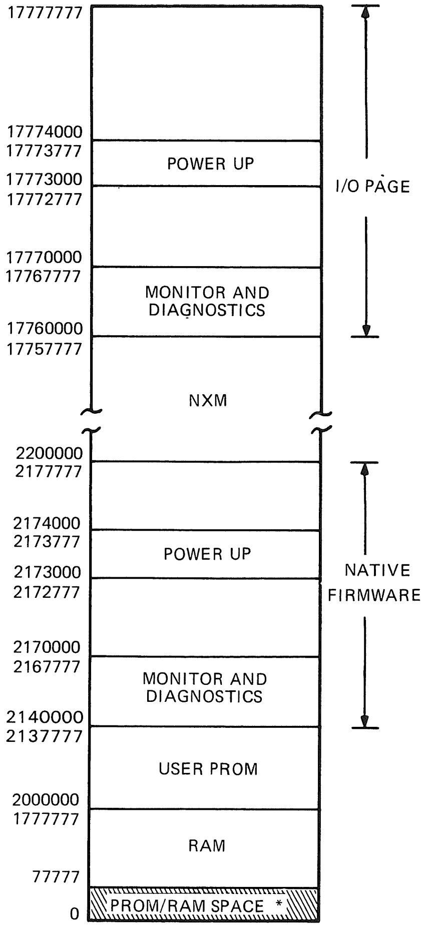
2.2.1 Boot/Self-test Switch

The boot/self-test switch is a 16 position switch that is used if the board is configured to execute firmware (rather than MicroODT) upon power-up. It has three functions.

1. It determines how the KXJ11-CA will act when a special interrupt condition exists, including whether or not self-tests will run (see Section 3.5).

2. It determines whether special interrupt handling is performed either by user code or by firmware.

3. It determines where in memory the on-board PROM is mapped. There are two alternatives -- low memory or high memory. The memory maps associated with low and high PROM mapping are shown in Figures 2-2 and 2-3, respectively.



ADDRESSES 77777-0 MATCH ADDRESSES 2077777-2000000

MR-17263

Figure 2-2 Memory Mapping - PROM in Low Memory

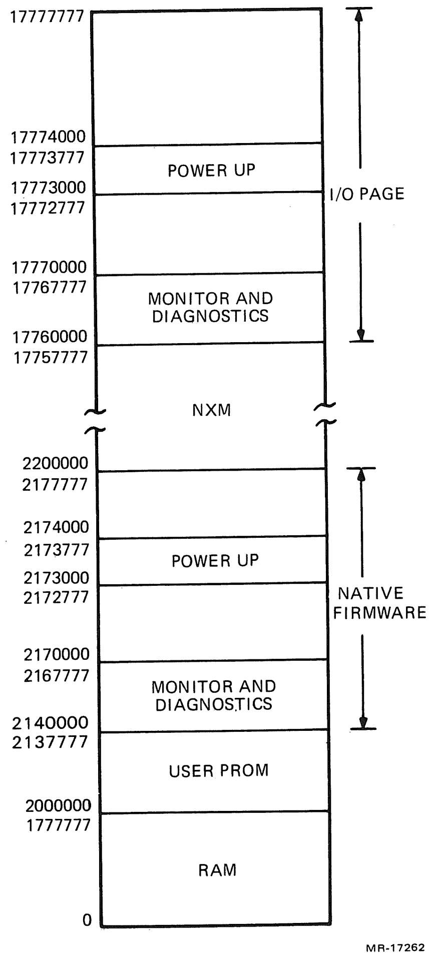


Figure 2-3 Memory Mapping - PROM in High Memory

The location of the boot/self-test switch is shown in Figure 2-4. Table 2-2 summarizes the functions associated with each switch position.

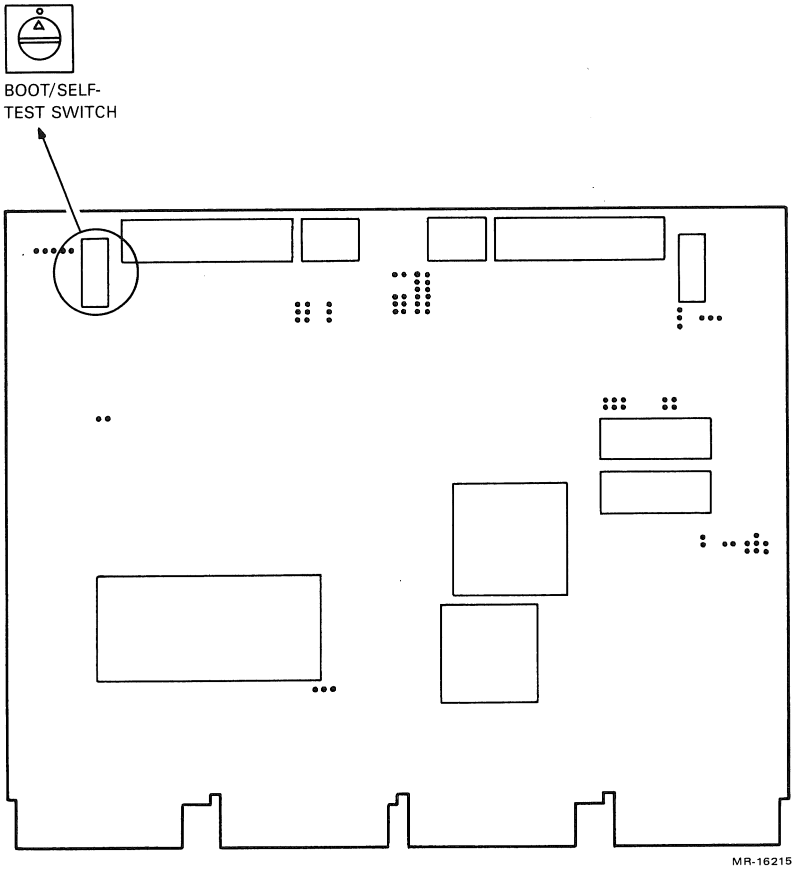


Figure 2-4 Boot/Self-test Switch

Table 2-2 Boot/Self-test Switch Functions

Switch Position	KXJ11-CA Special Interrupt Response	Special Interrupt Handling	PROM Mapping
0	User PROM application code is executed. No self-tests are performed.	Firmware	Low
1	User PROM application code is executed. Auto self-tests are performed.	Firmware	Low
2	User PROM application code is executed. Auto self-tests are performed. The user (P)ROM checksum test is also performed.	Firmware	Low
3	Application code is booted from a TU58 via SLU1. Auto self-tests are performed, then the TU58 primary bootstrap is executed.	Firmware	High
4	MicroODT is entered. No self-tests are performed.	Firmware	High
5	Auto self-tests are performed. The KXJ11-CA awaits command from the arbiter via TPR0.	Firmware	High
6	No self-tests are performed. The KXJ11-CA awaits a command from the arbiter via TPR0.	Firmware	High
7	Auto self-tests are performed continuously. No application code is booted or executed. Loopback connectors (see Section 2.5.3) are installed for these tests.	None	High
8	User PROM application code is executed. No self-tests are performed.	User Code	Low
9	User PROM application code is executed. Auto self-tests are performed.	User Code	Low
10	User PROM application code is executed. Auto self-tests are performed. The user (P)ROM checksum test is also performed.	User Code	Low

Table 2-2 Boot/Self-test Switch Functions (Cont)

Switch Position	KXJ11-CA Special Interrupt Response	Special Interrupt Handling	PROM Mapping
11	Application code is booted from a TU58 via SLU1. Auto self-tests are performed, then the TU58 primary bootstrap is executed.	User Code	High
12	MicroODT is entered. No self-tests are performed	User Code	High
13	Auto self-tests are performed. The KXJ11-CA awaits a command from the arbiter via TPR0.	User Code	High
14	No self-tests are performed. The KXJ11-CA awaits a command from the arbiter via TPR0.	User Code	High
15	Auto self-tests are performed continuously. No application code is booted or executed. Loopback connectors (see Section 2.5.3) are installed for these tests.	None	High

Notes:

1. Switch position 6 is the factory-shipped configuration.

2. The encoded value of the boot/self-test switch position is available in the KXJCSRB register in bits <7:4>. For example, switch position 1 would be encoded as 0001 in KXJCSRB <7:4>.

3. The user (P)ROM checksum test looks for a checksum at the highest word address of user (P)ROM. Similarly, the firmware checksum test looks for a checksum at the highest word address of the firmware PROM. Either checksum is calculated and checked according to the following DECPROM algorithm:

```
CHECKSUM = 0
FOR I = number of PROM addresses to be checksumed DO
      CHECKSUM = CHECKSUM + contents of address
      (high order carry from addition is discarded)
      CHECKSUM = ROTATE_LEFT_ONE_BIT
      (bit0 -> bit1, bit1 -> bit2, .... ,bit15 -> bit0)
NEXT I
```

4. Special interrupt handling can be performed by user code in switch positions 8-15. This function is useful in applications that need to continue running after the Q-Bus signal BHALT or the Q-Bus signal BINIT has been asserted. For switch positions 0 through 7, special interrupt handling is done by firmware.

Table 2-2 Boot/Self-test Switch Functions (Cont)

5. If the KXJ11-CA is in standalone mode, switch positions 5, 6, 13, and 14 should not be used. These postions cause the KXJ11-CA to idle and wait for a command. In standalone mode, the KXJ11-CA will idle indefinitely, waiting for an arbiter command that will never come.

2.2.2 Q-Bus Size

The KXJ11-CA may be configured to handle 16-, 18-, or 22-bit Q-Bus addressing. This is accomplished with the Q-Bus size jumpers (see Figure 2-5). 22-bit addressing is selected as part of the factory-shipped configuration.

```
Jumper Connection            Description

M3 o   o M5                  22-bit addressing selected*
   |   |
M4 o   o M6

M3 o   o M5                  18-bit addressing selected
       |
M4 o   o M6

M3 o   o M5                  16-bit addressing selected
   |
M4 o   o M6
```

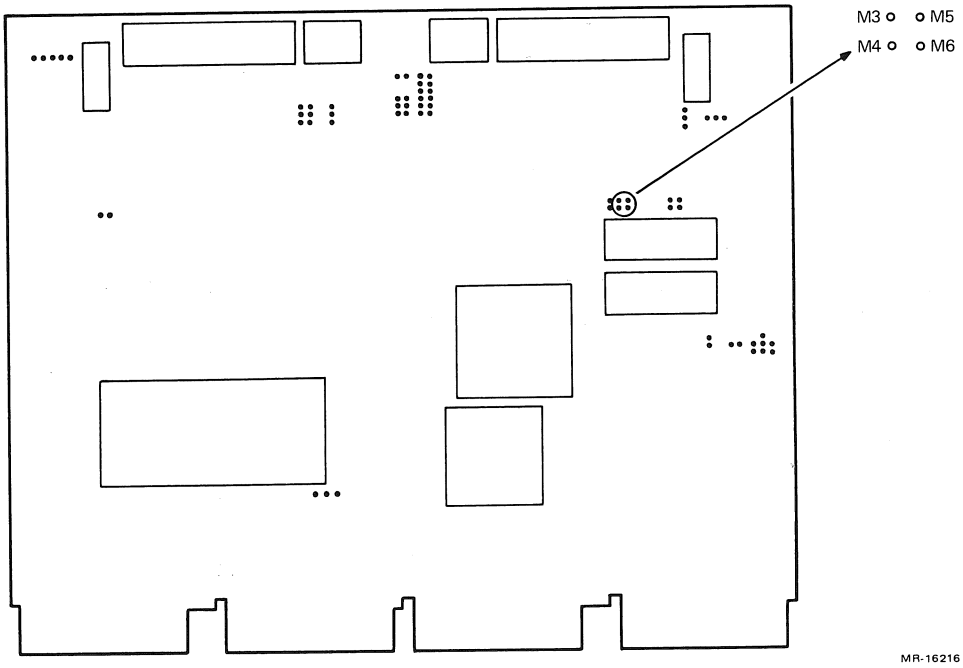


Figure 2-5 Q-Bus Size Selection

* Factory-shipped configuration

2.2.3 Q-Bus Base Address Selection

In systems with multiple I/O processor boards, make sure each board has a unique Q-Bus base address to distinguish the boards from one another. This is accomplished on the KXJ11-CA by setting the ID switch and installing or removing a jumper that connects M1 and M2.

Table 2-3 lists the base addresses that can be selected. Table 2-3 lists 22-bit addresses. If the KXJ11-CA is configured for 16- or 18-bit addressing, use the lower 16 or 18 bits of the addresses specified in Table 2-3.

Table 2-3 Q-Bus Base Address Selection

ID Switch Position	Base Address (Jumper IN)	Base Address (Jumper OUT)
0	*	*
1	*	*
2	17760100	17762100
3	17760140	17762140
4	17760200	17762200
5	17760240	17762240
6	17760300	17762300
7	17760340	17762340
8†	17775400†	17777400
9	17775440	17777440
10	17775500	17777500
11	17775540	17777540
12	17775600	17777600
13	17775640	17777640
14	17775700	17777700
15	17775740	17777740

* These switch positions disable the Q-Bus interface. That is, the KXJ11-CA is running in standalone mode.

† Factory-shipped configuration

Figure 2-6 shows the locations of jumper connections M1 and M2, and the ID switch. The factory-shipped base address is 17775400.

Jumper Connection	Description
○ M1 ○ M2	Factory-shipped configuration Base address = 17775400

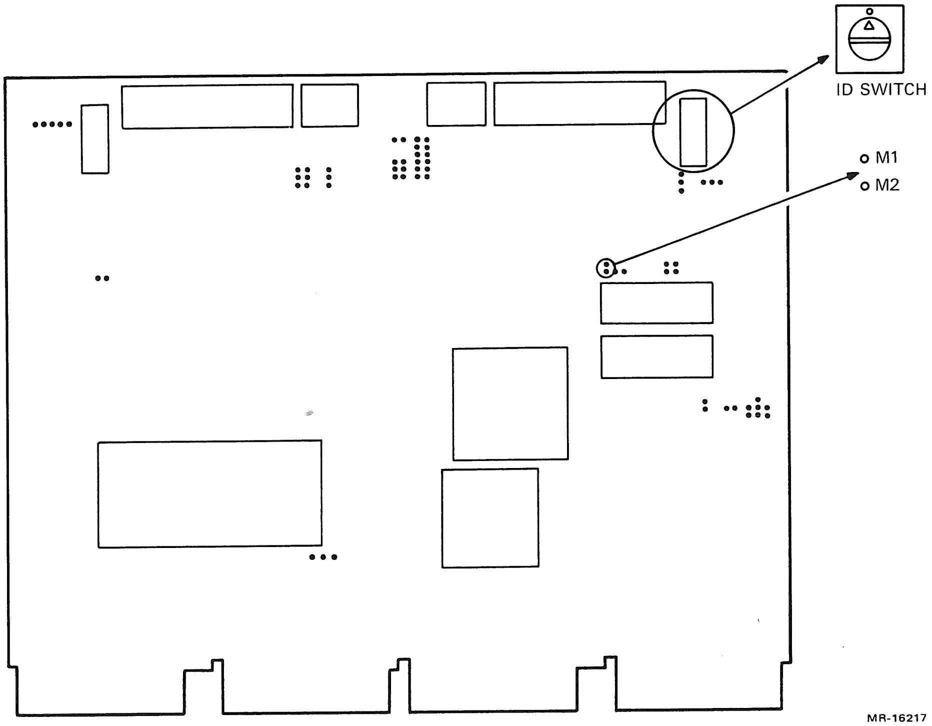


Figure 2-6 Q-Bus Base Address Selection

2.2.4 DMA Requests

DMA requests to the on-board DMA transfer controller (DTC) may come from several sources. The KXJ11-CA has a set of jumpers which enable or disable DMA requests from: (1) the SLU2 channel A receiver, (2) the SLU2 channel A transmitter, or (3) the on-board 8036 PIO counter/timer. The location of these jumpers is shown in Figure 2-7. Only two of the three sources may be specified (jumpered) at one time. The two sources that are jumpered as part of the factory-shipped configuration are SLU2 channel A receiver and SLU2 channel A transmitter.

Jumper Connection	Description
M11 M10 M9 M8 M7 o——o o o o	Allows DMA channel 0 requests from SLU2 channel A receiver*
M11 M10 M9 M8 M7 o o o——o o	Allows DMA channel 1 requests from PIO counter/timer
M11 M10 M9 M8 M7 o o o o——o	Allows DMA channel 1 requests from SLU2 channel A transmitter*

NOTE

Do not connect a jumper between connections M10 and M9. This configuration is not supported.

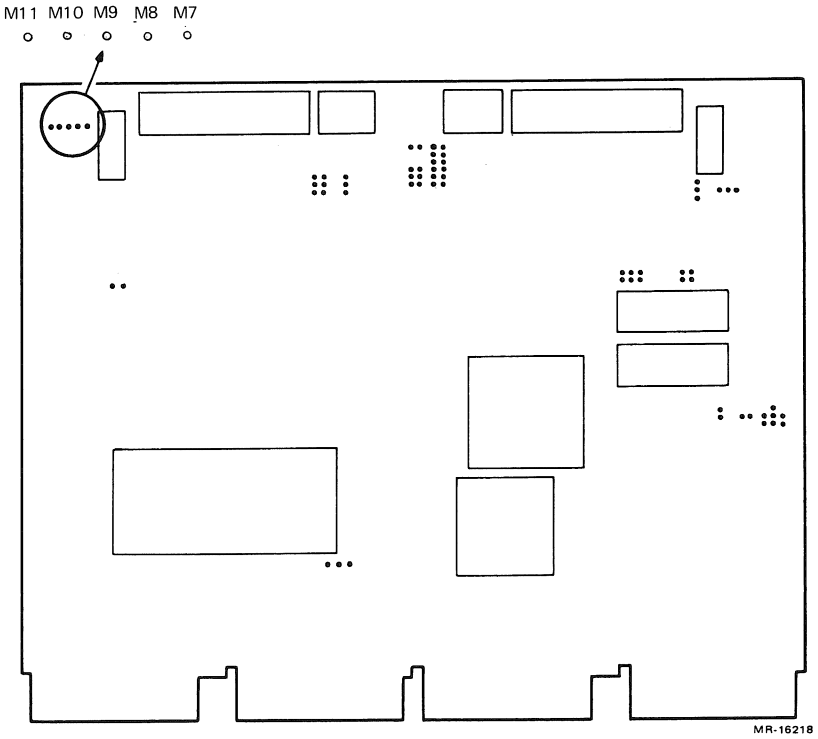


Figure 2-7 DMA Requests

* Factory-shipped configuration

2.2.5 Locked Instruction Enable

The KXJ11-CA has a set of jumpers which enable or disable the locking characteristic of the WRTLCK, TSTSET, and ASRB interlocked instructions. The location of the jumpers is shown in Figure 2-8. Locking is disabled as part of the factory-shipped configuration. For most applications, locking must be disabled. If locking is enabled, a Q-Bus timeout may cause a trap to location 4 if the Q-Bus is heavily loaded, and one of these instructions is executed.

Jumper Connection	Description
M64 M65 M66 o——o o	The locking characteristic of the WRTLCK, TSTSET, and ASRB instructions is enabled
M64 M65 M66 o o——o	The locking characteristic of the WRTLCK, TSTSET, and ASRB instructions is disabled*

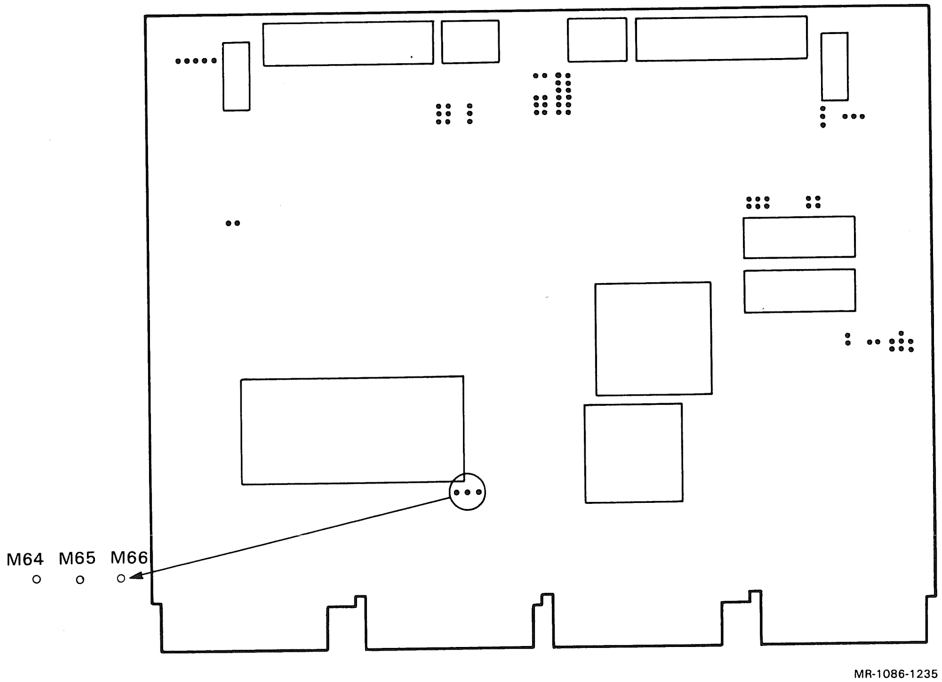


Figure 2-8 Locked Instruction Enable

* Factory-shipped configuration

2.2.6 BREAK Enable Selection

There is a jumper on the board that enables or disables console BREAK requests from SLU1 (the on-board DLART) to the J-11. The location of this jumper is shown in Figure 2-9. A BREAK is generated by SLU1 when a console terminal is attached to the system and the BREAK key on the console keyboard is pressed. When BREAK is received, the J-11 executes MicroODT. BREAK requests are enabled as part of the factory-shipped configuration.

Jumper Connection	Description
M13 o——o M12	Console BREAK requests enabled*
M13 o o M12	Console BREAK requests disabled

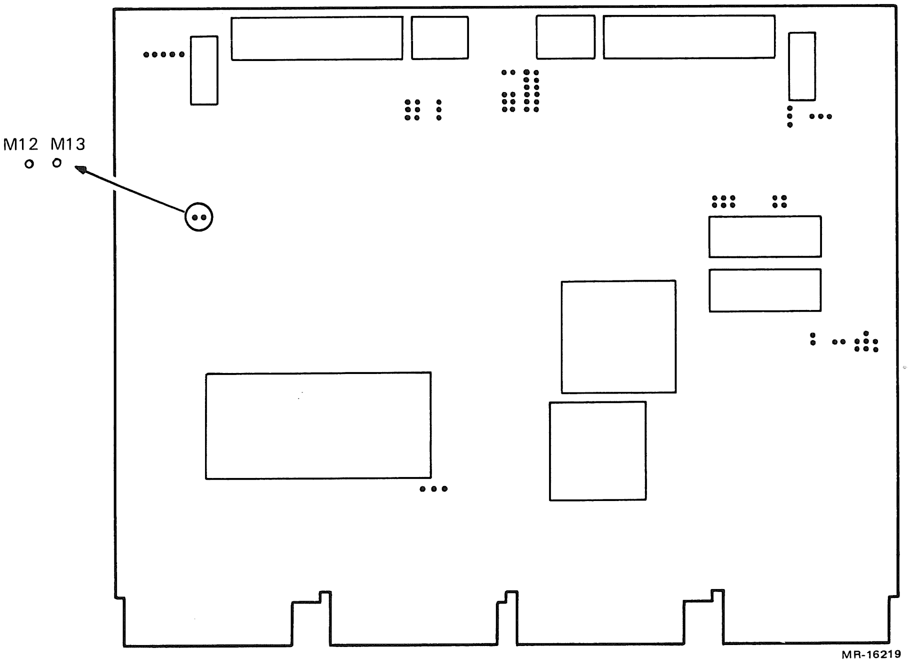


Figure 2-9 BREAK Enable

* Factory-shipped configuration

2.2.7 HALT Option Selection

A jumper on the KXJ11-CA determines what action will be taken if a HALT instruction is executed in kernel mode. The location of this jumper is shown in Figure 2-10. The jumper affects the state of bit 3 of the Maintenance Register (see Section 3.2.10). If the jumper is installed (the factory-shipped configuration), a HALT instruction executed in kernel mode causes the processor to enter MicroODT. If the jumper is not installed, the KXJ11-CA traps to location 4 in kernel instruction space and sets bit 7 in the CPU error register.

Jumper Connection	Description
M15 o——o M14	MicroODT is entered when a HALT instruction is executed in kernel mode*
M15 o o M14	KXJ11-CA traps to location 4 in kernel instruction space and sets bit 3 of the CPU error register if a HALT instruction is executed in kernel mode

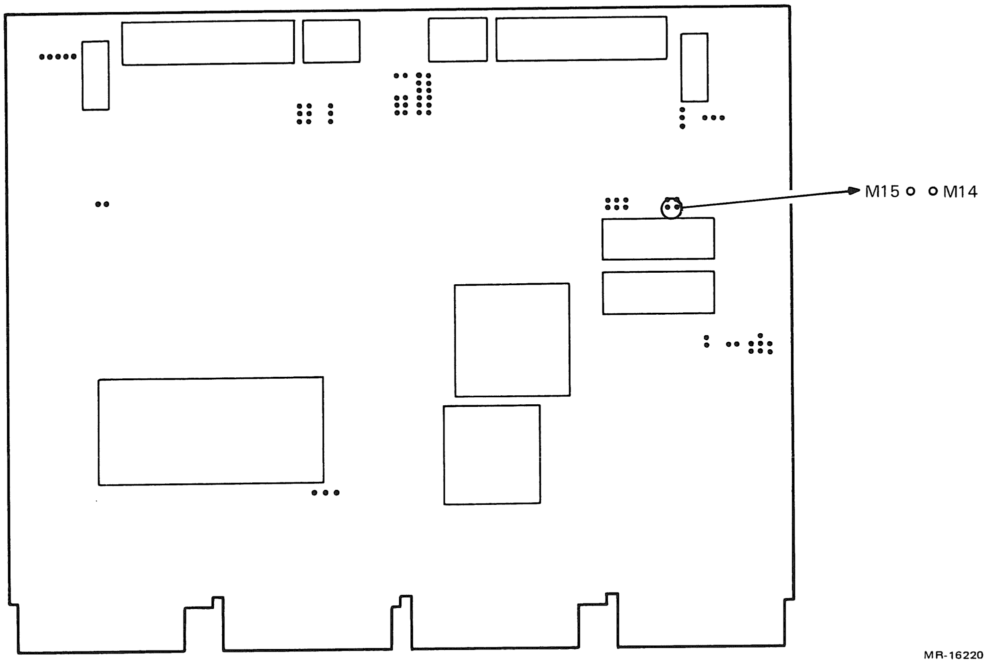


Figure 2-10 HALT Option Selection

* Factory-shipped configuration

2.2.8 Power-Up Option Selection

The power-up jumper (see Figure 2-11) determines what action the KXJ11-CA will take when the board is powered up or reset. The jumper affects the state of bit 2 of the Maintenance Register (see Section 3.2.10). At power-up, if the jumper is installed, the processor enters MicroODT with the PS register cleared. This is also known as power-up option 1. If the jumper is not installed, the (factory-shipped configuration), the KXJ11-CA executes the firmware power-up code at location 173000 during power-up (PC = 173000, PS = 340). This is also known as power-up option 3. Only power-up options 1 and 3 are used for the KXJ11-CA.

Jumper Connection	Description
M17 o——o M16	MicroODT is entered during power-up*
M17 o o M16	The KXJ11-CA bootstraps through location 173000 during power-up*

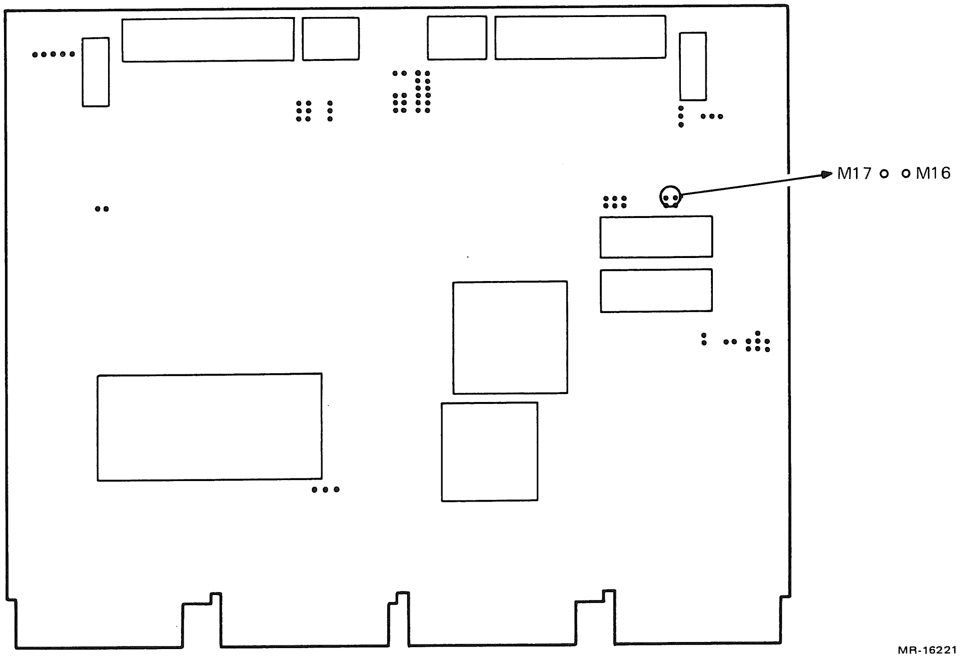


Figure 2-11 Power-Up Option Selection

* Factory-shipped configuration

2.2.9 PROM Addressing

The KXJ11-CA can be jumpered to accommodate various PROM types. The location of the PROM addressing jumper is shown in Figure 2-12. If the jumper is not installed, the on-board PROMs use 15-bit addresses. PROMS such as the Intel 2764 (8K X 8) and 27128 (16K X 8) use 15-bit addresses. If the jumper is installed, the PROMs use 16-bit addresses. This accommodates PROMs such as the Intel 27256 (32K X 8) that use 16-bit addresses. 15-bit PROM addressing is specified as part of the factory-shipped configuration.

Jumper Connection	Description
○ M19 ○ M18	15-bit addressing selected*
○ M19 │ ○ M18	16-bit addressing selected

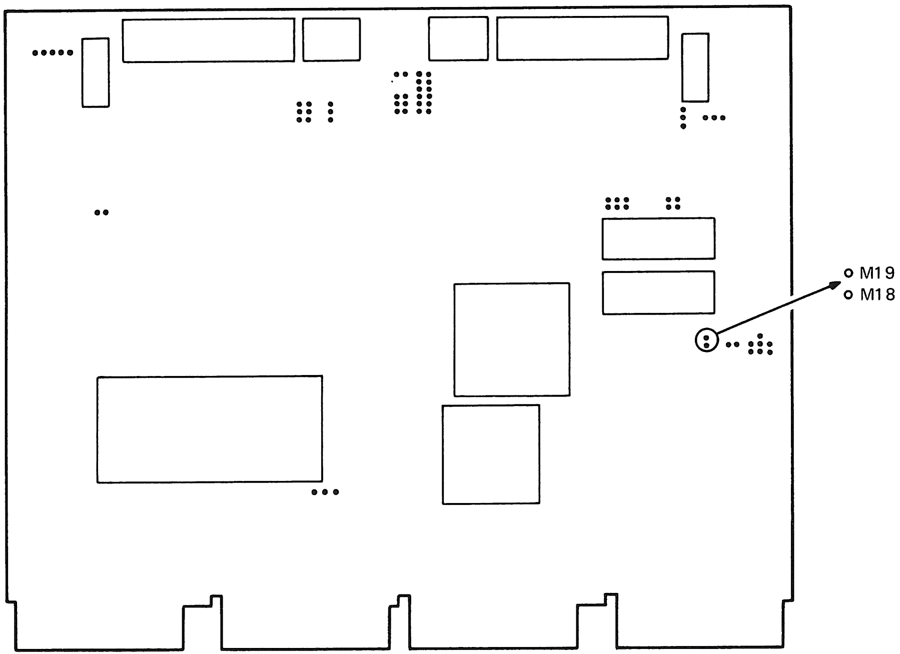


Figure 2-12 PROM Addressing

* Factory-shipped configuration

2.2.10 SLU1 Baud Rate

The jumpers shown in Figure 2-13 select the default baud rate for the SLU1 transmitter and receiver. The default baud rate for SLU1 is set when the KXJ11-CA is powered up or reinitialized. This rate can be changed under software control, if KXJCSRJ<3> is set. Table 2-4 shows the various baud rates that can be selected. A default baud rate of 9600 is specified as part of the factory-shipped configuration.

Jumper Connection	Description
M60 o——o M59	Factory-shipped configuration
M58 o o M57	9600 baud
M56 o——o M55	

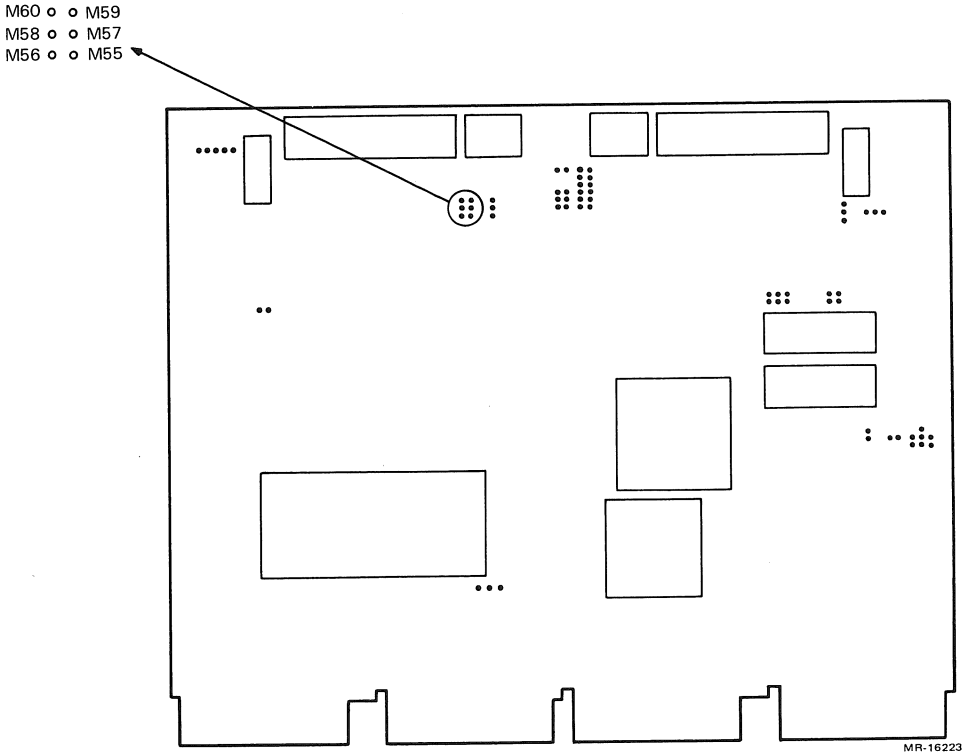


Figure 2-13 SLU1 Baud Rate

Table 2-4 SLU1 Baud Rate Jumpering

Baud Rate	M56 to M55	M58 to M57	M60 to M59
38400	In	In	In
19200	In	In	Out
9600*	In	Out	In
4800	In	Out	Out
2400	Out	In	In
1200	Out	In	Out
600	Out	Out	In
300	Out	Out	Out

* Factory-shipped configuration

2.2.11 SLU1 Transmitter

The SLU1 transmitter can be jumpered to send either single-ended (RS423) or differential (RS422) asynchronous serial data via connector J3. The location of the jumpers is shown in Figure 2-14. RS423 transmission is selected as part of the factory-shipped configuration.

Jumper Connection	Description
M63 M62 M61 ○ ○——○	RS423 transmission selected*
M63 M62 M61 ○——○ ○	RS422 transmission selected

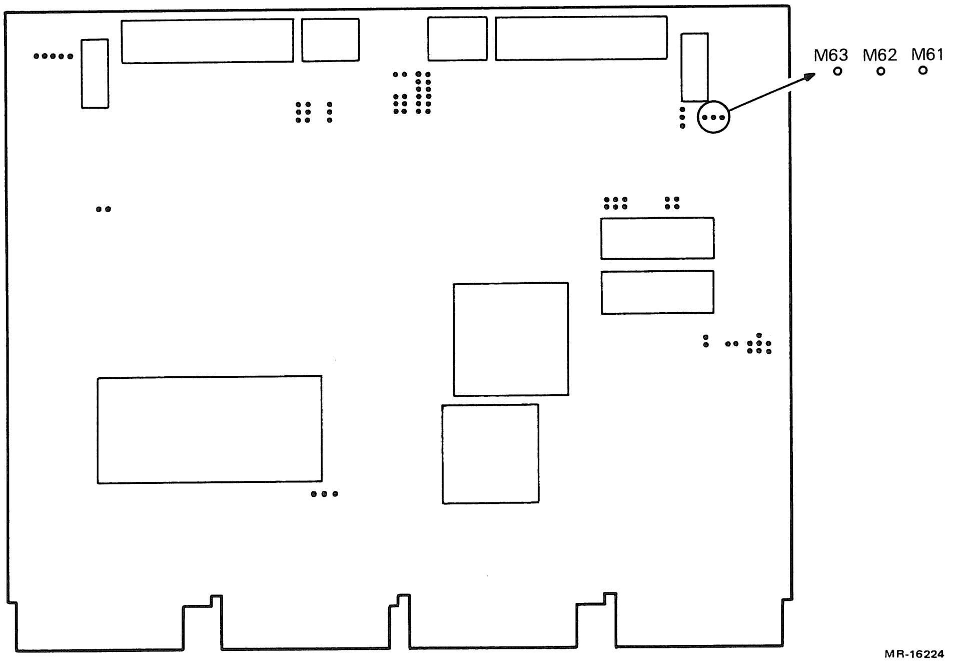


Figure 2-14 SLU1 Transmitter

* Factory-shipped configuration

2.2.12 SLU1 Receiver

The SLU1 receiver can be jumpered to receive either single-ended (RS423) or differential (RS422) asynchronous serial data via connector J3. The location of the jumper is shown in Figure 2-15. RS423 reception is selected as part of the factory-shipped configuration.

Jumper Connection	Description
M48 M47 o———o	RS422 reception selected
M48 M47 o o	RS423 reception selected*

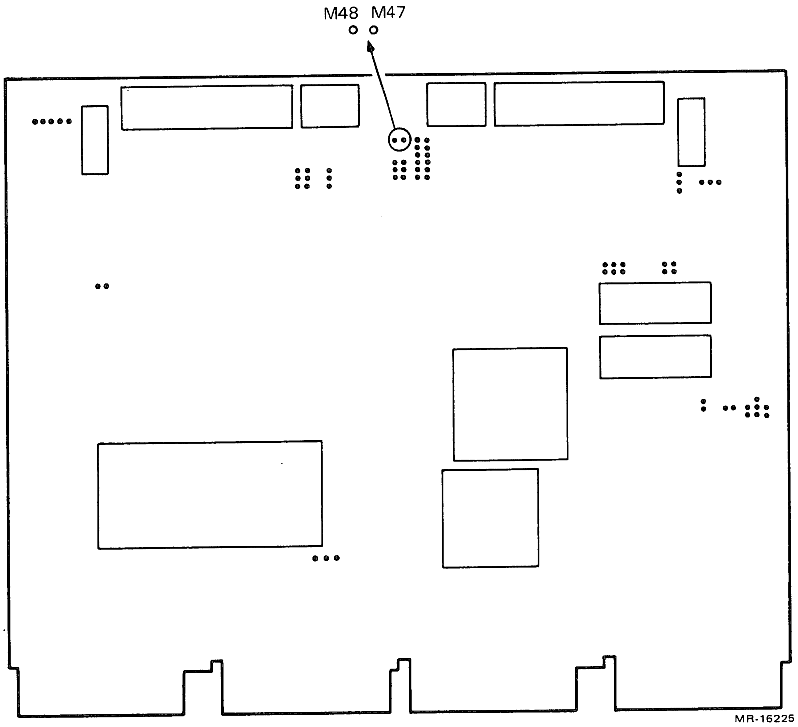


Figure 2-15 SLU1 Receiver

* Factory-shipped configuration

2.2.13 SLU2 Channel A Receiver

The SLU2 channel A receiver can be jumpered to receive either single-ended (RS423) or differential (RS422) serial data via connector J1. The location of the jumpers is shown in Figure 2-16. RS422 reception is selected as part of the factory-shipped configuration.

Jumper Connection	Description
M34 o——o M33	RS422 reception selected*
M32 o——o M31	
M30 o——o M29	
M28 o——o M27	
M26 o——o M25	
M24 o——o M23	
M34 o o M33	RS423 reception selected
M32 o o M31	
M30 o o M29	
M28 o o M27	
M26 o o M25	
M24 o o M23	

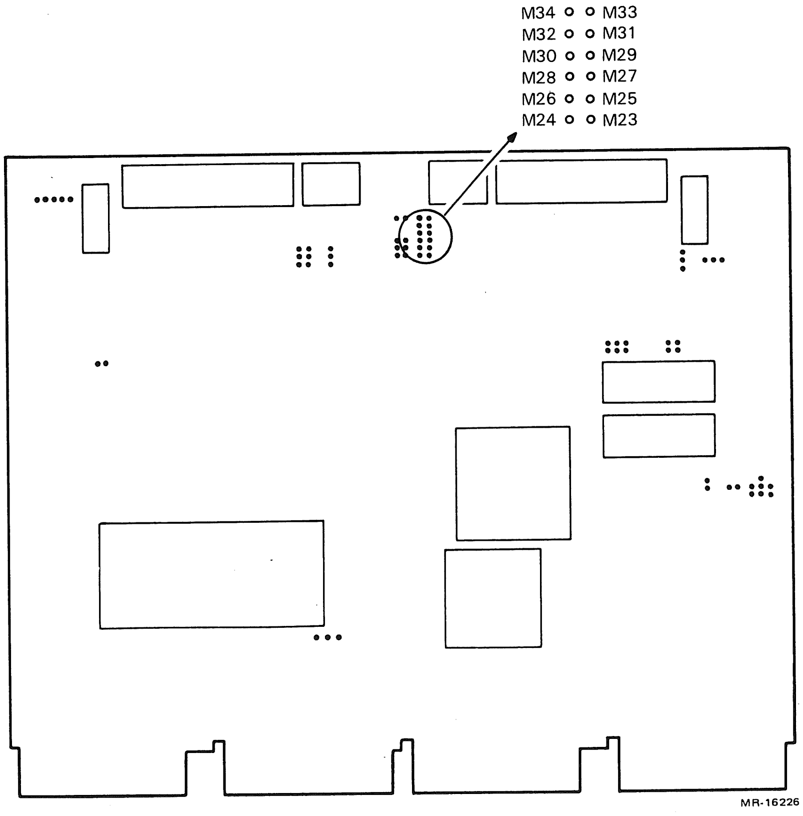


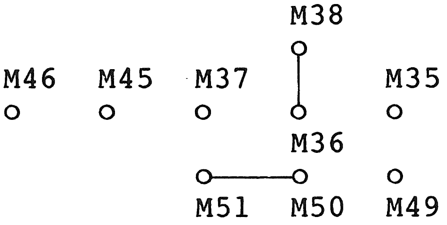
Figure 2-16 SLU2 Channel A Receiver

* Factory-shipped configuration

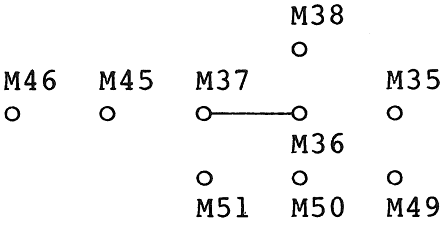
2.2.14 SLU2 Channel B Transmitter

The SLU2 channel B transmitter can be jumpered to send single-ended (RS423), differential (RS422), or party line (CCITT R1360) serial data via connector J2. The location of the jumpers is shown in Figure 2-17. RS422 transmission is selected as part of the factory-shipped configuration.

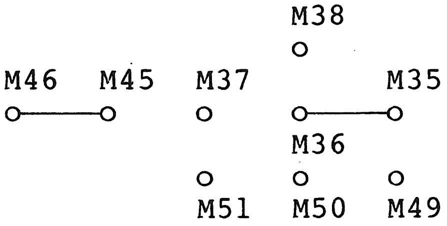
Jumper Connection **Description**



RS422 transmission selected*



RS423 transmission selected



Party line transmission selected

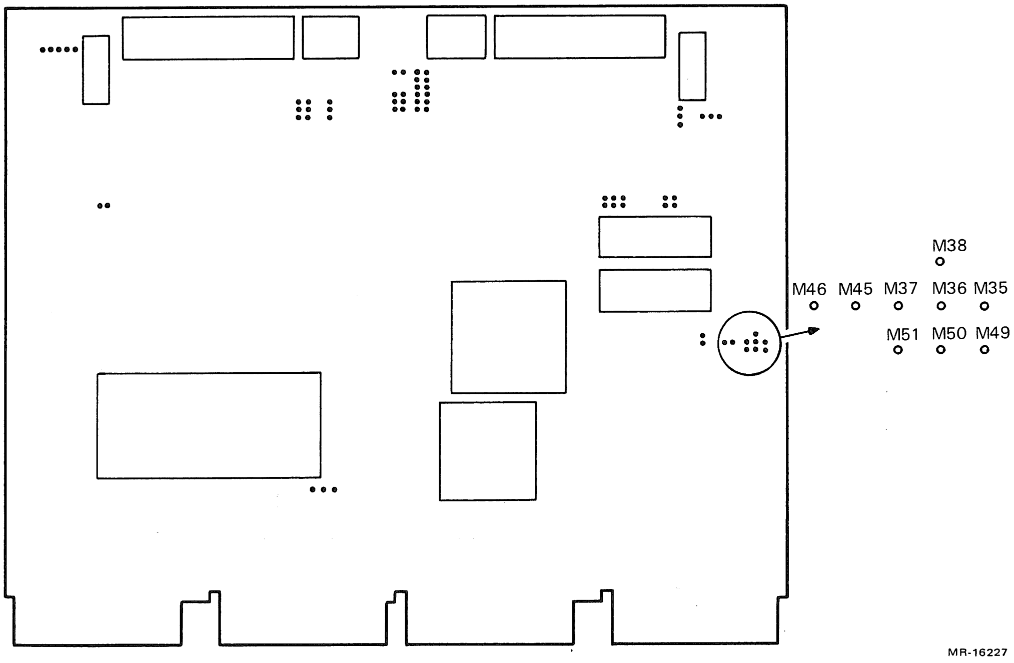


Figure 2-17 SLU2 Channel B Transmitter

* Factory-shipped configuration

2.2.15 SLU2 Channel B Receiver

The SLU2 channel B receiver can be jumpered to receive single-ended (RS423), differential (RS422), or party line (CCITT R1360) serial data via connector J2. Two groups of jumpers are involved and shown in Figure 2-18. RS422 reception is selected as part of the factory-shipped configuration.

```
Jumper Connection               Description

M44  o     o  M43  o  M20       RS422 reception selected*
                   |
M42  o-----o  M41  o  M21
M40  o-----o  M39  o  M22

M44  o     o  M43  o  M20       RS423 reception selected
                   |
M42  o     o  M41  o  M21
M40  o     o  M39  o  M22

M44  o-----o  M43  o  M20       Party line reception selected
M42  o     o  M41  o  M21
                   |
M40  o     o  M39  o  M22
```

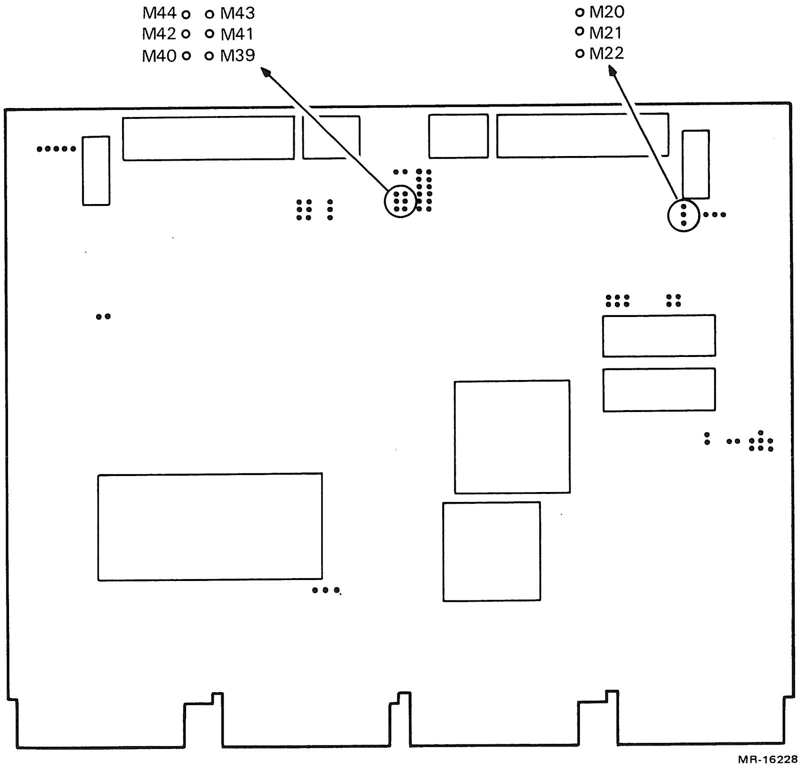


Figure 2-18 SLU2 Channel B Receiver

* Factory-shipped configuration

2.2.16 Real-Time Clock Interrupt

SLU1 (the on-board DLART) can generate real-time clock interrupts at frequencies of 50 and 60 Hz. Jumpers M52, M53, and M54 select either the 50 Hz or the 60 Hz real-time clock as input to the interrupt control logic. If interrupts are enabled, each clock "tick" results in a maskable priority level 6 interrupt request to the on-board J-11. The location of the real-time clock interrupt jumpers is shown in Figure 2-19. A real-time clock rate of 60 Hz is specified as part of the factory-shipped configuration.

Jumper Connection		Description
o	M54	60 Hz real-time clock selected*
o	M53	
o	M52	
o	M54	50 Hz real-time clock selected
o	M53	
o	M52	

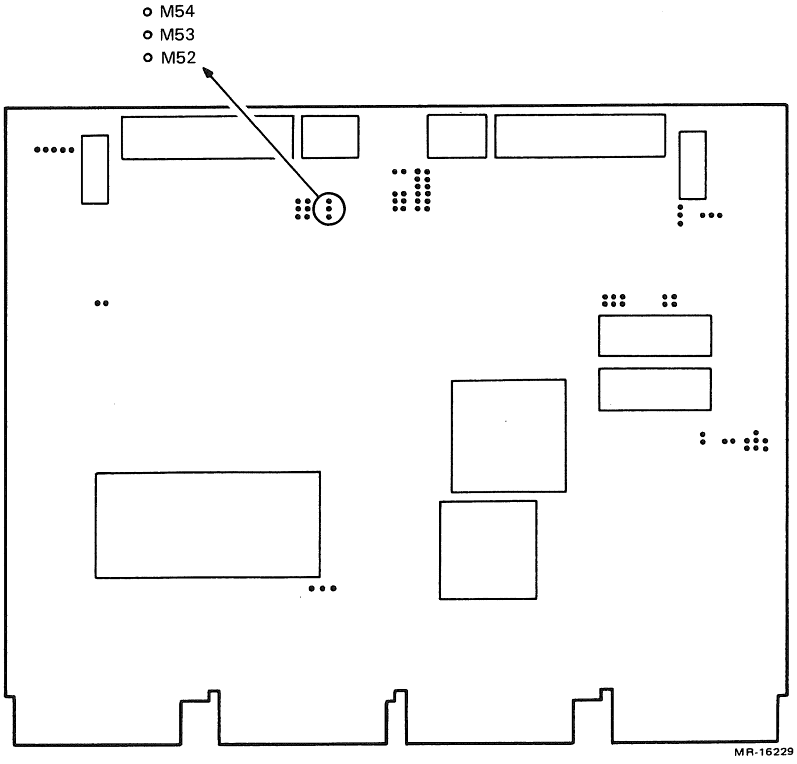


Figure 2-19 Real-Time Clock Interrupt

* Factory-shipped configuration

2.3 POWER SUPPLY CONSIDERATIONS

When installing the KXJ11-CA, make sure the power supply can handle the extra load presented by the board. The KXJ11-CA draws a maximum of 6A at +5V. In addition, the KXJ11-CA draws a maximum of 1.4A at +12V, for systems with the DLV11-KA option, or .4A maximum at +12V, for systems without the DLV11-KA option. The board adds 2.7 ac loads and 1.0 dc loads to the bus.

In standalone mode, at least four power fingers (backplane connections) and four ground fingers for +5 Vdc must be connected to the power supply. In addition, at least two power fingers and two ground fingers for +12 Vdc must be connected to the power supply.

2.4 INSTALLING THE KXJ11-CA INTO A BACKPLANE

The KXJ11-CA plugs into any DEC standard quad height Q-Bus backplane (see Figure 2-20). No special backplane wiring or jumpering is required to accommodate the KXJ11-CA. However, the grant structure must be preserved if there are blank slots between the KXJ11-CA and the top of the backplane. This can be accomplished by inserting grant cards where appropriate (Figure 2-21 is an example of the use of grant cards.) The dual height grant card (M8659) preserves grant continuity for slots A and B, and grant card G7272 preserves both the DMA and interrupt grant continuity for slot C. The KXJ11-CA board must also be configured for the proper Q-Bus address size.

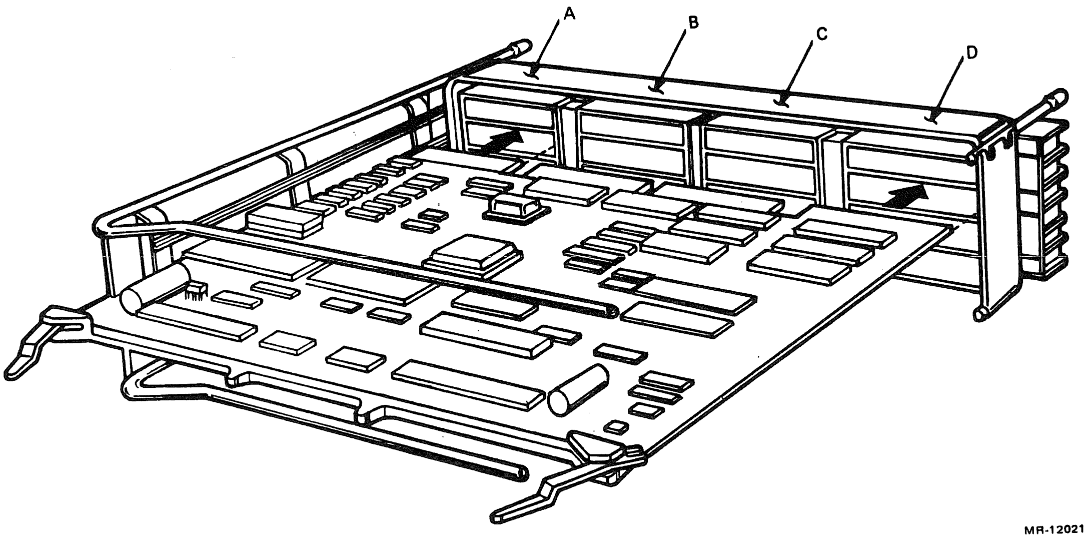


Figure 2-20 Backplane Installation

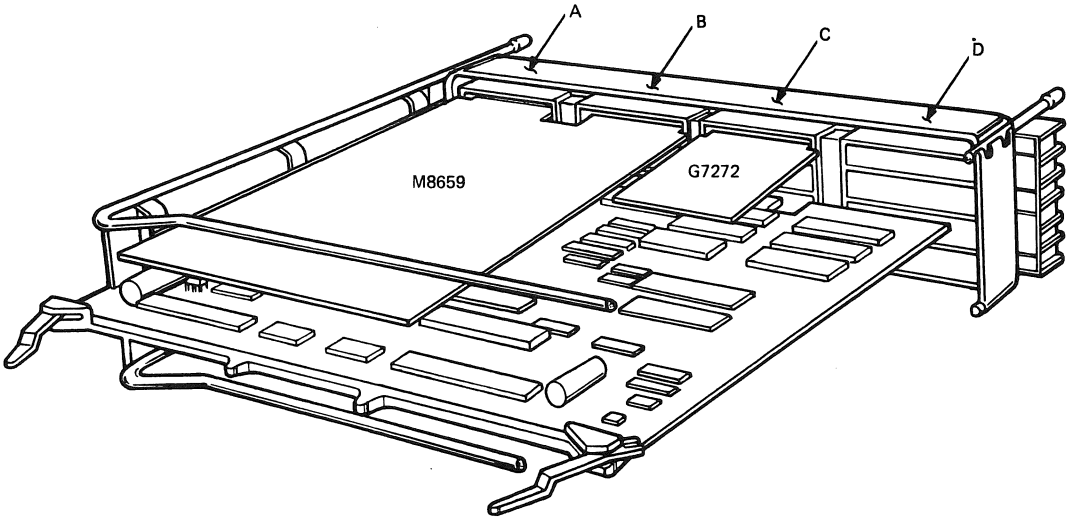


Figure 2-21   Using Grant Cards

2.4.1 Edge Connector Pin Assignments

Table 2-5 summarizes the edge connector pin assignments for the KXJ11-CA. The board is designed to mate with DEC standard quad height backplanes for Q-Bus-based systems.

Table 2-5 KXJ11-CA Pin Identification

Component Side		Solder Side	
	KXJ11-CA		KXJ11-CA
Pin	Signal	Pin	Signal
AA1	NC	AA2	+5V
AB1	NC	AB2	NC
AC1	BDAL16 L	AC2	GND
AD1	BDAL17 L	AD2	NC
AE1	NC	AE2	BDOUT L
AF1	NC	AF2	BRPLY L
AH1	NC	AH2	BDIN L
AJ1	GND	AJ2	BSYNC L
AK1	NC	AK2	BWTBT L
AL1	NC	AL2	BIRQ4 L
AM1	GND	AM2	BIAKI L
AN1	BDMR L	AN2	BIAKO L
AP1	BHALT L	AP2	BBS7 L
AR1	NC	AR2	BDMGI L
AS1	NC	AS2	BDMGO L
AT1	GND	AT2	BINIT L
AU1	NC	AU2	BDAL0 L
AV1	+5VB	AV2	BDAL1 L
BA1	BDCOK H	BA2	+5V
BB1	BPOK H	BB2	NC
BC1	BDAL18 L	BC2	GND
BD1	BDAL19 L	BD2	+12V
BE1	BDAL20 L	BE2	BDAL2 L
BF1	BDAL21 L	BF2	BDAL3 L
BH1	NC	BH2	BDAL4 L
BJ1	GND	BJ2	BDAL5 L
BK1	NC	BK2	BDAL6 L
BL1	NC	BL2	BDAL7 L
BM1	GND	BM2	BDAL8 L
BN1	BSACK L	BN2	BDAL9 L
BP1	NC	BP2	BDAL10 L
BR1	NC	BR2	BDAL11 L
BS1	NC	BS2	BDAL12 L
BT1	GND	BT2	BDAL13 L
BU1	NC	BU2	BDAL14 L
BV1	+5V	BV2	BDAL15 L

Table 2-5 KXJ11-CA Pin Identification (Cont)

Component Side		Solder Side	
Pin	KXJ11-CA Signal	Pin	KXJ11-CA Signal
CA1	NC	CA2	+5V
CB1	NC	CB2	NC
CC1	NC	CC2	GND
CD1	NC	CD2	NC
CE1	NC	CE2	NC
CF1	NC	CF2	NC
CH1	NC	CH2	NC
CJ1	NC	CJ2	NC
CK1	NC	CK2	NC
CL1	NC	CL2	NC
CM1	NC	CM2	IAK L (Note 2)
CN1	NC	CN2	IAK L (Note 2)
CP1	NC	CP2	NC
CR1	NC	CR2	DMG L (Note 3)
CS1	NC	CS2	DMG L (Note 3)
CT1	GND	CT2	NC
CU1	NC	CU2	NC
CV1	NC	CV2	NC
DA1	NC	DA2	+5V
DB1	NC	DB2	NC
DC1	NC	DC2	GND
DD1	NC	DD2	NC
DE1	NC	DE2	NC
DF1	NC	DF2	NC
DH1	NC	DH2	NC
DJ1	NC	DJ2	NC
DK1	NC	DK2	NC
DL1	NC	DL2	NC
DM1	NC	DM2	NC
DN1	NC	DN2	NC
DP1	NC	DP2	NC
DR1	NC	DR2	NC
DS1	NC	DS2	NC
DT1	GND	DT2	NC
DU1	NC	DU2	NC
DV1	NC	DV2	NC

Notes:

1. NC = Not connected
2. Pin CM2 is jumpered to pin CN2 for the interrupt acknowledge daisy chain.
3. Pin CR2 is jumpered to pin CS2 for the DMA grant daisy chain.

2.5 CONNECTORS AND EXTERNAL CABLING

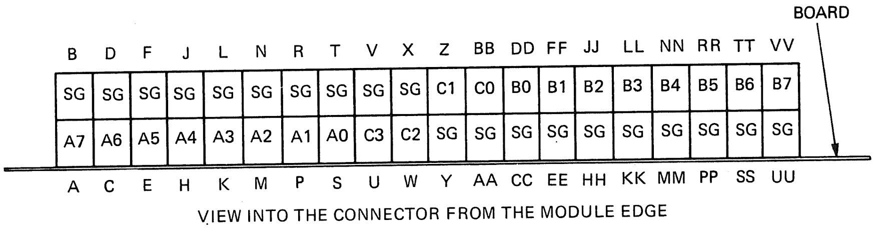
The KXJ11-CA communicates with external devices via a parallel I/O connector (J4) and three serial I/O connectors (J1, J2, and J3). This section specifies the pin assignments of these connectors and lists the types of cables that can be used with each connector.

2.5.1 Parallel I/O Interface (J4)

The parallel I/O (PIO) interface signals appear at connector J4. These signals are buffered. They can be driven over a 50 ft. distance via a ribbon cable, or round cable with a 40-pin AMP contact housing (AMP part number 746473-9) at each end. A PIO cable is not provided with the KXJ11-CA. The following PIO cables are recommended for use in the KXJ11 and are available from Digital Equipment Corporation.

BC06	R shielded ribbon cable
BC05	L "mirror image" cable

Figure 2-22 lists the pin assignments for J4, the parallel I/O connector.



B	D	F	J	L	N	R	T	V	X	Z	BB	DD	FF	JJ	LL	NN	RR	TT	VV
SG	SG	SG	SG	SG	SG	SG	SG	SG	SG	C1	C0	B0	B1	B2	B3	B4	B5	B6	B7
A7	A6	A5	A4	A3	A2	A1	A0	C3	C2	SG	SG	SG	SG	SG	SG	SG	SG	SG	SG
A	C	E	H	K	M	P	S	U	W	Y	AA	CC	EE	HH	KK	MM	PP	SS	UU

VIEW INTO THE CONNECTOR FROM THE MODULE EDGE

NOTE: SG = SIGNAL GROUND
A0 - A7 = PORT A
B0 - B7 = PORT B
C0 - C3 = PORT C

MR-12615

Figure 2-22 Parallel I/O Interface Pin Assignments

2.5.2 Serial I/O Lines (J1, J2, J3)

The KXJ11-CA has three serial I/O lines.

- SLU2 channel A (J1), a synchronous/asynchronous serial line with modem control
- SLU2 channel B (J2), a synchronous/asynchronous serial line without modem control
- SLU1 (J3), the console asynchronous serial line (no modem control)

Each serial line is compatible with the EIA RS232-C and RS422/RS423 protocols. In addition, SLU2 channel B (J2) is compatible with the CCITT R1360 party line protocol. Interfacing the KXJ11-CA with a 4-20 mA current loop device via the serial lines can be done by using the DLV11-KA option.

Serial line cables must be supplied by the user. The following cables are recommended for use for the J2 and J3 serial I/O lines and are available from Digital Equipment Corporation.

BC20N-05 A 5-foot EIA RS232-C null modem cable for a direct connection between the KXJ11-CA and an EIA terminal. This cable has a 10-pin (2 x 5) AMP female connector on one end and a 25-pin RS232-C female connector on the other.

BC21B-05 A 5-foot EIA RS232-C modem cable for a connection between the KXJ11-CA and a modem or acoustic coupler. This cable has a 10-pin (2 x 5) AMP female connector on one end and a 25-pin RS232-C male connector on the other.

BC20M-50 A 50-foot EIA RS422 or RS423 cable for a direct connection between the KXJ11-CA and a remote processor. Used in applications requiring high data transmission speeds (up to 19.2 K baud), this cable has a 10-pin (2 X 5) AMP female connector on each end.

The pin designations for J2 and J3 are shown in Figure 2-23.

All three serial lines are factory configured to handle differential inputs and outputs. If you change the configuration of any of the serial lines to handle single-ended inputs or outputs, make sure the return (-) signal(s) on the cable are tied to signal ground.

Because there is no standard cable available from Digital Equipment Corporation for SLU2 channel A, you need to construct your own cable. A standard 40-pin AMP connector may be used (AMP part number 746473-9) for connection to J1. Figure 2-24 illustrates the pin assignments for SLU2 channel A (J1).

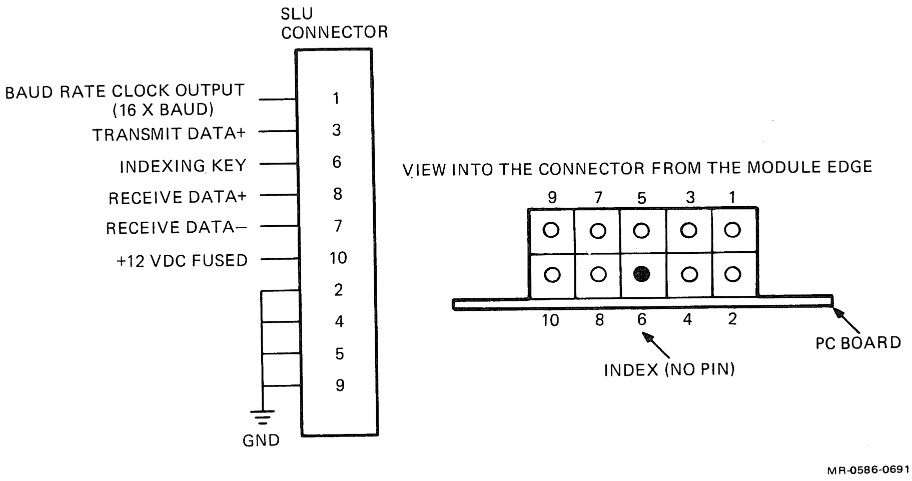


Figure 2-23 J2 and J3 Pin Assignments (10-Pin)

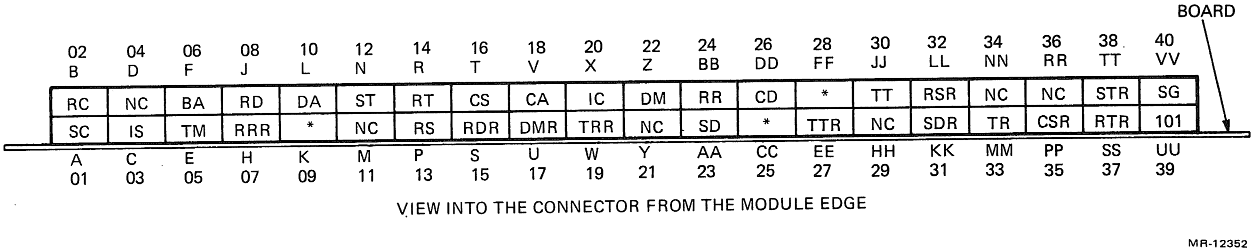

BOARD
02 04 06 08 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40
B D F J L N R T V X Z BB DD FF JJ LL NN RR TT VV
RC NC BA RD DA ST RT CS CA IC DM RR CD * TT RSR NC NC STR SG
SC IS TM RRR * NC RS RDR DMR TRR NC SD * TTR NC SDR TR CSR RTR 101
A C E H K M P S U W Y AA CC EE HH KK MM PP SS UU
01 03 05 07 09 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
VIEW INTO THE CONNECTOR FROM THE MODULE EDGE
MR-12352

Figure 2-24 J1 Pin Assignments (40-Pin)

Tables 2-6 through 2-8 show the correspondence between the pins of the standard connectors for the RS422/RS423, RS232, and CCITT protocols, and the pins of J1. These tables make it easy to construct an appropriate cable. The KXJ11-CA register address associated with each signal is specified in the last column of each table for ease of programmer reference. The register descriptions in Chapters 3 and 6 provide further details.

Table 2-6 RS422/RS423 Interface to J1

Pin	Circuit	Direction	Function	RS-232	CCITT	Pin	Location
1	SHIELD	-	Protective Ground			-	
2	SI	From Modem		CI	112	5,E	17777522
3	SPARE					-	
4	SD	To Modem	Send Data (+)	BA	103	23,AA 6,F	17775706
5	ST	From Modem	Send Timing (+)	DB	114	12,N	17777520
6	RD	From Modem	Receive Data (+)	BB	104	8,J	17775702
7	RS	To Modem	Request to Send (+)	CA	105	13,P 18,V	17775704
8	RT	From Modem	Receive Timing (+)	DD	115	14,R	17775720
9	CS	From Modem	Clear to Send (+)	CB	106	16,T	17775700
10	LL	To Modem	Local Loop		141	25,CC Gen.	Dummy
11	DM	From Modem	Data Mode (+)	CC	107	22,Z	17775710
12	TR	To Modem	Terminal Ready (+)	CD	108/2	33,M 26,DD	17777520
13	RR	From Modem	Receiver Ready (+)	CF	109	24,BB	17775700
14	RL	To Modem	Remote Loop		140	9,K Gen.	Dummy
15	IC	From Modem	Incoming Call	CE	125	20,X	17775710
16	SF/SR	To Modem	Select Frequency Signal Rate Select	 CH	126 111	3,C 3,C	17777520
17	TT	To Modem	Terminal Timing (+)	DA	113	30,JJ 10,L	17777530
18	TM	From Modem	Test Mode		142	5,E	17777522
19	SG	To Modem	Signal Ground	AB	102	40,W	
20	RC	From Modem	Receive Common		102b	2,B	
21	SPARE					-	
22	SDR	To Modem	Send Data (-)			31,KK	
23	STR	From Modem	Send Timing (-)			38,TT	
24	RDR	From Modem	Receive Data (-)			15,S	
25	RSR	To Modem	Request to Send (-)			32,LL	
26	RTR	From Modem	Receive Timing (-)			37,SS	
27	CSR	From Modem	Clear to Send (-)			35,PP	
28	IS	To Modem	Terminal in Service			3,C	17777520
29	DMR	From Modem	Data Mode (-)			17,V	
30	TRR	To Modem	Terminal Ready (-)			19,W	
31	RRR	From Modem	Receiver Ready (-)			7,H	
32	SS	To Modem	Select Standby		116	28,FF Gen.	Dummy
33	SQ	From Modem	Signal Quality	CG	110	-	
34	NS	To Modem	New Signal			-	
35	TTR	To Modem	Terminal Timing (-)			27,EE	

Table 2-6 RS422/RS423 Interface to J1 (Cont)

Pin	Circuit	Direction	Function	RS-232	CCITT	Pin	Location
36	SB	From Modem	Standby Indication		117	-	
37	SC	To Modem	Send Common		102a	1,A	

Notes on Figure 2-23 and Table 2-6:

1. Pins K 9, 25 CC, and 28 FF are driven by dummy generators that disable RL (CCITT 140), LL (CCITT 141), and SS (CCITT 116) respectively.

2. The label NC indicates no connection.

3. The suffix R in a three-letter pin label (such as RDR) signifies that the pin is associated with the return side of a differential driver or receiver.

4. Circuit IS can be redefined to mean SF. Or IS can be redefined as SR. In the second case, TM is also redefined as SI.

Table 2-7 RS232-C Interface to J1

Pin	Circuit	Direction	Function	CCITT	Pin	Location
1	AA	-	Protective Ground	101	39,UU	
2	BA	To Modem	Transmitted Data	103	6,F	17775706
3	BB	From Modem	Received Data	104	8,J	17775702
4	CA	To Modem	Request to Send	105	18,V	17775704
5	CB	From Modem	Clear to Send	106	16,T	17775700
6	CC	From Modem	Data Set Ready	107	22,Z	17775710
7	AB	-	Signal Ground	102	40,W	
8	CF	From Modem	Receiver Ready	109	24,BB	17775700
9	-	(From Modem)	(+ DC Test Voltage)		-	
10	-	(To Modem)	(- DC Test Voltage)		-	
11	-	-	Unassigned		-	
12	SCF	From Modem	Secondary Carrier Detector	122	-	
13	SCB	From Modem	Secondary Clear to Send	121	-	
14	SBA	To Modem	Secondary Transmitted Data	118	-	
15	DB	From Modem	Transmitter Clock	114	12,N	17777520
16	SBB	From Modem	Secondary Received Data	119	-	
17	DD	From Modem	Receiver Clock	115	14,R	17777520
18	-	To Modem	Receiver Dibit Clock			
19	SCA	To Modem	Secondary Request to Send	120	-	
20	CD	To Modem	Data Terminal Ready	108/2	26,DD	17777520

Table 2-7 RS232-C Interface to J1 (Cont)

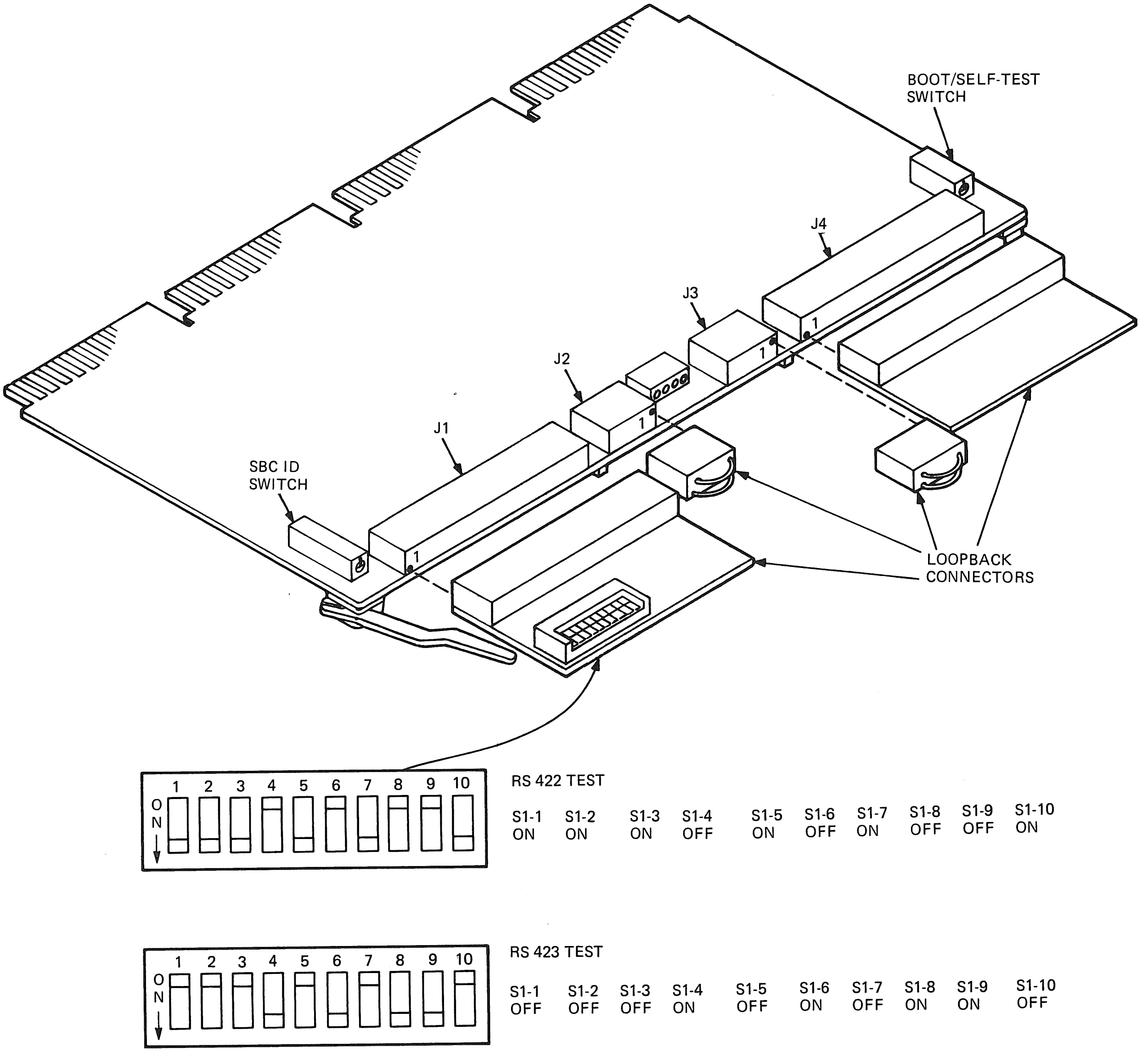
Pin	Circuit	Direction	Function	CCITT	Pin	Location
21	CG	From Modem	Signal Quality Detector	110	-	
22	CE	From Modem	Ring Indicator	125	20,X	17775710
23	CH/CI	To Modem	Data Rate Selector	111	5,E	17777522
				112	3,C	17777520
24	DA	To Modem	External Transmitter Clock	113	10,L	17777530
25	CN	To Modem	Force Busy		-	

Table 2-8 CCITT/V.35 Interface to J1

Pin	Circuit	Direction	Function	RS232	RS449	Pin	Location
A	101	-	Protective Ground	AA		39,UU	
B	102	-	Signal Ground	AB	SG	40,W	
C	105	To Modem	Request to Send	CA	RS	18,V	17775704
D	106	From Modem	Ready for Sending	CB	CS	16,T	17775700
E	107	From Modem	Data Set Ready	CC	DM	22,Z	17775710
F	109	From Modem	RCV Line Signal Det	CF	RR	24,BB	17775700
H	108/1	To Modem	Connect Data Set				
	108/2	To Modem	Data Terminal Ready	CD	TR	26,DD	17777520
J	125	From Modem	Calling Indicator	CE	IC	20,X	17775710
R	104	From Modem	Received Data A	BB	RD	8,J	17775702
T	104	From Modem	Received Data B		RD	-	
V	115	From Modem	Receive Timing A	DD	RT	14,R	17777520
X	115	From Modem	Receive Timing B		RT	-	
Y	114	From Modem	Transmit Timing A	DB	ST	12,N	17777520
AA	114	From Modem	Transmit Timing B		ST	-	
P	103	To Modem	Transmit Data A	BA	SD	6,F	17775706
S	103	To Modem	Transmit Data B		SD	-	
U	113	To Modem	Terminal Timing A	DA	TT	10,L	17777530
W	113	To Modem	Terminal Timing B		TT	-	

2.5.3 Loopback Connectors

Loopback connectors (not provided with the KXJ11-CA) are attached to the serial or parallel communication ports to determine whether or not they are operating correctly (see Figure 2-25). They are typically used in conjunction with the running of diagnostic programs, and in some firmware self-tests (see Sections 2.2.1). These connectors may be ordered from Digital Equipment Corporation or may be built by the user.



RS 422 TEST

S1-1	S1-2	S1-3	S1-4	S1-5	S1-6	S1-7	S1-8	S1-9	S1-10
ON	ON	ON	OFF	ON	OFF	ON	OFF	OFF	ON

RS 423 TEST

S1-1	S1-2	S1-3	S1-4	S1-5	S1-6	S1-7	S1-8	S1-9	S1-10
OFF	OFF	OFF	ON	OFF	ON	OFF	ON	ON	OFF

Figure 2-25   Loopback Connectors

There are three different types of loopback connectors available from Digital. A 10-pin loopback connector (DEC part number H3270) is plugged into either J2, to test SLU2 channel B, or into J3 to test SLU1. A 40-pin loopback connector (DEC part number H3022) is plugged into J1 to test SLU2 channel A. This loopback connector can also be configured to test RS422 or RS423 operation (see Figure 2-25). The third type of loopback connector is also 40 pins (DEC part number H3021), and is plugged into J4 to test the parallel I/O port.

2.6 ERROR DETECTION AND REPORTING WITH THE LEDS

There are four LEDs on the edge of the KXJ11-CA board which the native firmware uses to indicate the state of the board. These are especially useful for diagnostic purposes during power-up or reinitialization. These LEDs verify that the board is operating properly or, if there is a problem with the board, can help the user locate the difficulty. Table 2-9 summarizes the conditions the LEDs can indicate.

During power-up or reinitialization, all four LEDs are illuminated for approximately 1/2 second if they are working properly. When the KXJ11-CA is installed in a backplane in a box, the LEDs are labeled L4 through L1 from left to right. If the KXJ11-CA runs its self-tests (this is determined by the setting of the boot/self-test switch), L4 is off and L3 - L1 should be on as the self-tests run. If one of the self-tests fails, L4 is illuminated and L3 - L1 indicate the test that failed. Self-tests are run in the order listed in Table 2-9. Thus, if a test fails, the user can also determine which tests (if any) passed.

If all the self-tests run without error, the KXJ11-CA performs a boot operation. The boot/self-test switch setting determines which function is performed. L4 remains off and L3 - L1 indicate the status of the executing code. Note that the boot/self-test switch may be set so self-tests are not run. If self-tests have not been run, then L4 is off and L3 - L1 indicate the state of the board as it executes code.

Table 2-9 LED Display Definitions

	LEDs			
L4	L3	L2	L1	Meaning
x	x	x	x	All LEDs on for 1/2 sec. at the start of a power-up or reinitialization operation
x	x	x	x	Can't access Control/Status Registers in I/O page (fatal error) OR the power-up jumper (M16-M17) is installed, which precludes self-tests (in Micro ODT).
x	o	o	o	DMA or RTC test failed
x	o	o	x	RAM test failed
x	o	x	o	ROM checksum test failed
x	o	x	x	Serial line test of SLU1 failed
x	x	o	o	Serial line test of SLU2 channel A failed
x	x	o	x	Serial line test of SLU2 channel B failed

Table 2-9   LED Display Definitions (Cont)

LEDs L4	 L3	 L2	 L1	 Meaning
x	x	x	o	Parallel port test failed
o	x	x	x	Auto self-tests running. Auto self-tests do not require loopback connectors.
o	x	x	o	Loopback tests and auto self-tests running
o	x	o	x	Q-Bus ODT mode
o	x	o	o	Unused
o	o	x	x	Waiting for command
o	o	x	o	Performing DTC load
o	o	o	x	TU58 primary bootstrap executing
o	o	o	o	Executing non-native code

Quick LED Reference

LEDs L4	 L3	 L2	 L1	 Meaning
x	-	-	-	Self-test error detected
x	x	x	x	Fatal self-test error detected
o	-	-	-	No self-test errors detected
o	o	o	o	Application running without error

Legend
x = On
o = Off
- = Don't care (either On or Off)

2.7 DIAGNOSTIC TESTING WITH XXDP+

The KXJ11-CA can be tested by running XXDP+, a diagnostic operating system that is booted from the user's system disk. This section explains how to run the XXDP+ diagnostics to test the KXJ11-CA. More information on XXDP+ is found in the XXDP+ System User's Manual (AC-F348F-MC).

When you have successfully booted XXDP+ from the system disk, a message such as the one shown below appears on the console terminal. The items that are blank (underscore) indicate values that are system dependent.

```
BOOTING UP XXDP-_______ MONITOR

XXDP-SM _____ MONITOR VERSION ____
BOOTED FROM ____
____KW OF MEMORY
NON-UNIBUS SYSTEM

RESTART ADDR: ______
THIS IS XXDP= ___ TYPE "H" or "H/L" FOR HELP
.
```

When the "period" prompt appears, the user types in

```
R ZKXA??<CR>
```

This initiates the running of the tests. The message

```
ZKA__.BIN
```

appears on the console, followed by several lines of system information (the underscore indicates characters that are system dependent). Then the following message should appear.

```
USE <ESC> TO HALT

KXJ FUNCTIONAL TEST

SWR     OCTAL      FUNCTION
---     -----      --------
15      100000     HALT ON ERROR
14      040000     INHIBIT ERROR SUMMARY
13      020000     INHIBIT ERROR REPORTS
12      010000     IOP IO# KNOWN GOOD FOR TESTING
11      004000     TEST STAND ALONE IOP
10      002000     ENABLE EXTENDED MEMORY TESTS
09      001000     LOOP ON ERROR
08      000400     LOOP ON TEST IN SWR<6:0>
07      000200     INHIBIT TEST NUMBER/TITLE

SWR = 140000  NEW =
```

At this point, type <CR>, which runs the tests until an error is detected. As the tests run, their results are displayed on the console. If an error is detected, a self-explanatory error message is displayed, and the tests halt if bit 15 in the SWR is set to 1. The halt causes an entry into MicroODT. To continue after an error has caused a halt, type

```
    P<CR>
```

using the console keyboard.

If no errors are detected, testing can be terminated by pressing the ESCAPE key (which halts the KXJ11-CA (and causes microODT to be entered) if the BREAK enable jumper is installed, or by pressing the BREAK key (which halts the arbiter).

CHAPTER 3
ARCHITECTURE

3.1 INTRODUCTION

This chapter describes the architecture of the KXJ11-CA and explains the operation of the various user-accessible portions of the KXJ11-CA.

In the case of the on-board I/O devices (chips), more detailed information is found in the technical manuals included in this documentation package. This chapter describes how the I/O devices operate on the KXJ11-CA. The information in the chapter may differ from the information in the technical manuals, since the technical manuals only describe the operations on a chip level. The differences, where they exist, are noted.

3.2 KXJ11-CA BLOCK DIAGRAM

Figure 3-1 illustrates the major operational elements and data paths of the KXJ11-CA. The sections that follow describe the important characteristics of these components.

3.2.1 J-11 Microprocessor

The J-11 microprocessor operates at 14.0 Mhz. It contains a full PDP-11 memory management unit (MMU) and executes the PDP-11 Extended Instruction Set (EIS). The processor also contains microdiagnostics and console ODT. Cache memory and the Floating Point Accelerator (FPA) are not included as part of the KXJ11-CA architecture. The start address is fixed at 173000 and the restart address is set at 173004. Status bits are used to determine the reason for a restart. J-11 power-up options 1 and 3 can be selected using jumper M17-M16.

3.2.2 RAM

The KXJ11-CA employs 256K X 1 dynamic RAM chips, and a 512 KB X 18 bit array is provided for memory and parity storage. RAM may be accessed locally or may be configured as shared memory (accessible both locally and from the Q-Bus) in quantities of 0 KB to 512 KB. Shared memory is assignable in 8 KB blocks on 8 KB address boundaries, and if more than one block is configured, all blocks are contigious. The memory is configured with the KXJCSRF and KXJCSRH registers, and is enabled by a bit in the KXJCSRJ register. KXJCSRF contains the starting Q-Bus address, as well as the number of 8 KB blocks assigned to the shared memory. KXJCSRH contains the ending Q-Bus address assigned to the shared memory. Section 3.8 provides more details on shared memory and how to set it up.

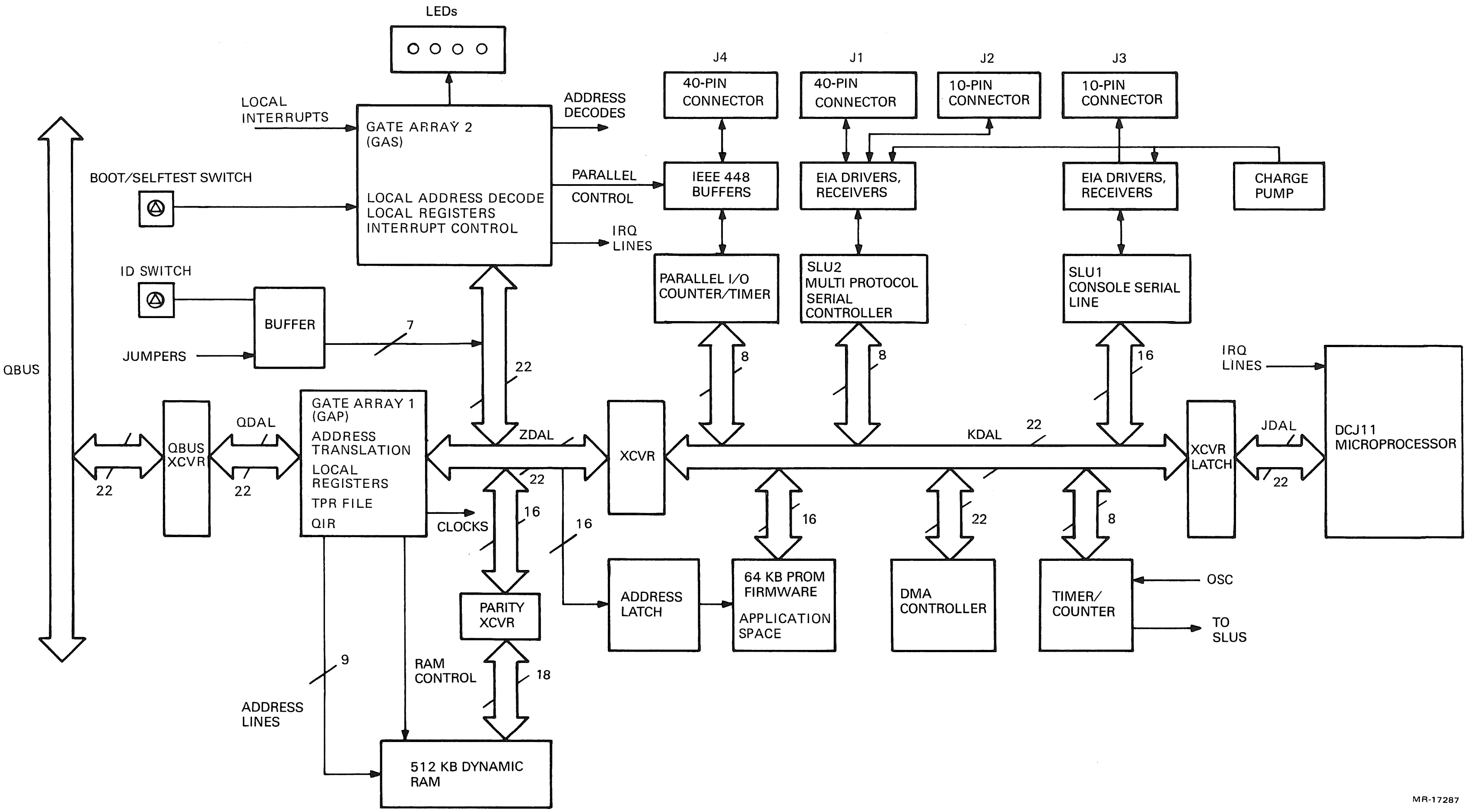
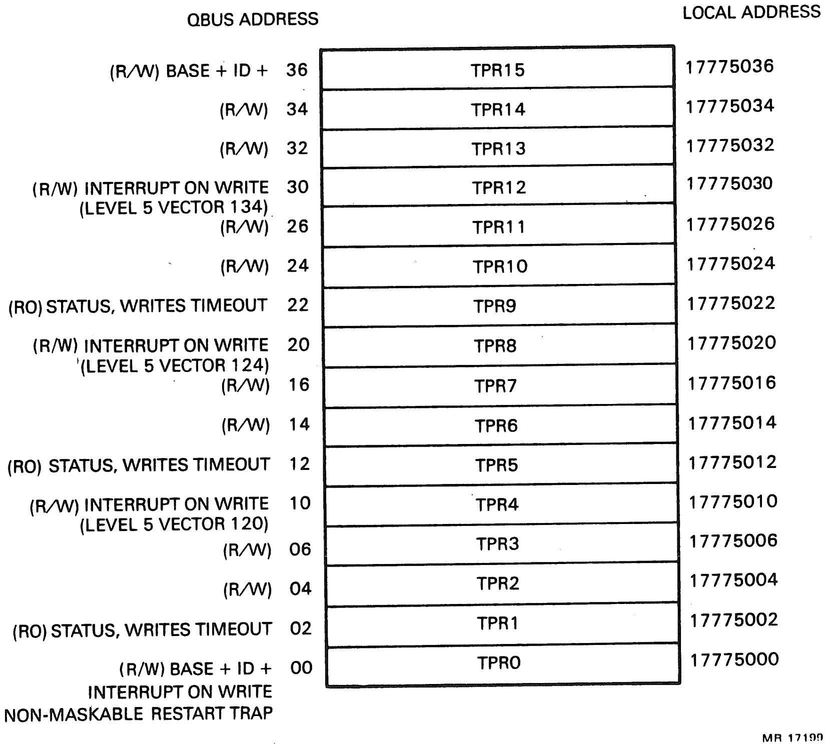


Figure 3-1 KXJ11-CA Block Diagram

3.2.3 Two Port Register (TPR) File

The Two Port Register (TPR) File is a 16 word set of registers that can be accessed either by the on-board J-11 microprocessor or by the Q-Bus. The TPR file is the primary means by which the Q-Bus arbiter controls and communicates with the KXJ11-CA. There are four groups of TPRs. TPR0 through TPR3 is a communication channel between the KXJ11-CA native firmware and the arbiter. The other three groups, TPR4 through TPR7, TPR8 through TPR11, and TPR12 through TPR15, typically act as communication channels between the user's application and the arbiter. All TPRs reside in the GAP on-board gate array (DC7036B). The TPRs are enabled by a bit in the KXJCSRD register. If TPRs are disabled, all TPRs except TPR0 are read-only from the arbiter side and always read as zeros. Writes to any register except TPR0 will time out if the TPRs are disabled. Writing to TPR0 with the TPRs disabled succeeds but reads a zero. This allows writing to TPR0<14> to cause a hardware reset. Figure 3-2 illustrates the TPR file.



QBUS ADDRESS			LOCAL ADDRESS
(R/W) BASE + ID +	36	TPR15	17775036
(R/W)	34	TPR14	17775034
(R/W)	32	TPR13	17775032
(R/W) INTERRUPT ON WRITE (LEVEL 5 VECTOR 134)	30	TPR12	17775030
(R/W)	26	TPR11	17775026
(R/W)	24	TPR10	17775024
(RO) STATUS, WRITES TIMEOUT	22	TPR9	17775022
(R/W) INTERRUPT ON WRITE (LEVEL 5 VECTOR 124)	20	TPR8	17775020
(R/W)	16	TPR7	17775016
(R/W)	14	TPR6	17775014
(RO) STATUS, WRITES TIMEOUT	12	TPR5	17775012
(R/W) INTERRUPT ON WRITE (LEVEL 5 VECTOR 120)	10	TPR4	17775010
(R/W)	06	TPR3	17775006
(R/W)	04	TPR2	17775004
(RO) STATUS, WRITES TIMEOUT	02	TPR1	17775002
(R/W) BASE + ID + INTERRUPT ON WRITE NON-MASKABLE RESTART TRAP	00	TPR0	17775000

Figure 3-2 Two Port Register (TPR) File

3.2.3.1 Arbiter/TPR Communication Protocol -- The following general protocol is implemented by KXJ11-CA firmware. It is used to handle arbiter commands when the arbiter is communicating with the KXJ11-CA registers TPR0 through TPR4. If the user wants the firmware to be ready to receive a command from the arbiter during power up, the boot/self-test switch should be set to positions 5 or 6.

1. TPR0 (the command register) is tested to determine if it is ready to receive a command. The KXJ11-CA is ready to receive a command only when all bits in TPR0 are zero.

2. If the command requires parameters, the arbiter loads them into TPR2 and/or TPR3.

3. The arbiter writes the command into TPR0.

4. The arbiter waits for TPR0 to be zero, indicating the completion of the execution of the command by the firmware.

5. The arbiter can check for any errors that may have occurred during command execution, by examining the appropriate bits in TPR1.

3.2.3.2 TPR0 -- From the Q-Bus, TPR0 can be interpreted or used in three different ways: as a KXJ11-CA control register, as a test register, or as a Q-Bus ODT register.

NOTE
This description of TPR0 assumes that bit 6 of KXJCSRJ (NMI ENable) is set.

If bit 15 is cleared when TPR0 is written from the Q-Bus, TPR0 is interpreted as a control register. If bit 15 is set when TPR0 is first accessed from the Q-Bus, TPR0 is interpreted as a test register. After a "Start Q-Bus ODT" command is issued (that is, when bit 3 is set while TPR0 is used as a control register), TPR0 is interpreted as a Q-Bus ODT command register until an "Exit Q-Bus ODT", "Proceed", or "Start Program" command is issued (that is, until bit 15, bit 4, or bit 3 is set). The sections that follow provide bit descriptions for all three interpretations of TPR0.

Bit 14 of TPR0 is always used as a "hard reset", which when set from the Q-Bus, causes a KXJ11-CA initialization that is handled by the KXJ11-CA native firmware. A hardware or software reset or a Q-Bus ODT GO command disables non-maskable interrupts.

To avoid unpredictable results, the user should not alter the TPRs used to pass parameters while a command or test is executing. The bit descriptions in the sections that follow specify the TPRs are used to pass parameters for the various commands and tests.

After any command, test, or Q-Bus ODT operation is executed (with or without error), TPR0 is cleared.

3.2.3.2.1 TPR0 as a Control Register -- If TPR0 is used as a control register (Figure 3-3), a set bit in TPR0<9:0> specifies a command. Only one command at a time can be specified. If any parameters accompany a command, they are passed through TPR2 and TPR3.

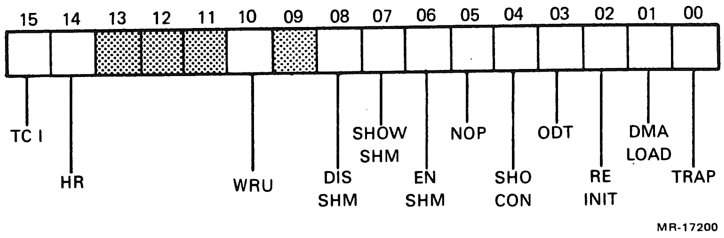


Figure 3-3 TPR0 as a Control Register

Bits	Name	Description
15	TC I	Test/Control Indicator - When set, TPR0 is used as a test register. When cleared, TPR0 is used as a control register.
14	HR	Hard reset - When set, a local power-up sequence occurs during the write portion of the current Q-Bus cycle. The setting of this bit cancels any previously invoked operations. Setting this bit causes a hardware reset and is equivalent to powering up the board. This occurs regardless of whether or not the native firmware is installed, or if the TPRs or non-maskable interrupts are enabled.
13:11		Not used (read/write)
10	WRU	What are you - When set, causes the firmware to write a value of 1 in TPR2, indicating that the board is a KXJ11-CA.
9		Not used (read/write)
8	DIS SHM	Disable shared memory - When set, disables shared memory. When set, and when shared memory is already disabled, sets bit 15 of TPR1 to indicate a command error.
7	SHOW SHM	Show shared memory - When set, bits <22:13> of the starting address of shared memory are loaded into TPR2 bits <9:0>. The number of pages to be shared minus one is loaded into TPR3. When set and when shared memory is disabled, sets bit 15 of TPR1.

Bits	Name	Description
6	EN SHM	Enable shared memory - When set, enables shared memory. Bits <21:13> of the Q-Bus starting address are taken from TPR2. The number of pages to be shared minus one are taken from TPR3.
5	NOP	No operation - This bit is reserved for use by Digital Equipment Corporation. It currently has no effect on KXJ11-CA operation.
4	SHO CON	Show configuration - When set, loads the boot/self-test switch setting into TPR3<7:4>. Also, writes TPR3<2:1> with the type of ROMs used on the board are shown below. **TPR<2:1>** **ROMs** 00 8 K X 8 01 16 K X 8 10 32 K X 8 11 Not used
3	ODT	Start Q-Bus ODT - When set, forces the KXJ11-CA into Q-Bus ODT mode. TPR0 is redefined (see Section 3.2.3.2.3) until bit 15 (EXIT), bit 3 (GO), or bit 4 (PROCEED) is set.
2	RE INIT	Restart/Initialize - When set, forces the native firmware to perform its power-up sequence. If TPR3 contains an 8 (decimal), the boot/self-test switch setting is used to determine which operations to perform. If TPR3 contains 0 through 7 (decimal), that value is used instead of the boot/self-test switch setting. TPR3 values greater than 8 (decimal) are not valid.
1	DMA LOAD	DMA load - When set, starts a chain load of the DTC. TPR3 is used to pass a "segment tag" parameter, and TPR2 is used to pass an "offset tag" parameter of a chain control table (see Section 4.3.1). After the operation is complete, bit 14 of TPR1 is set and the contents of the DTC Status Register are written into TPR2.

Bits	Name	Description
0	TRAP	Trap - When set, causes a trap emulation. The trap vector is in TPR2 (which contains the PC) and TPR3 (which contains the PSW). The trap vector is assumed to be in kernel I space.

3.2.3.2.2 TPR0 as a Test Register -- If TPR0 is used as a test register (Figure 3-4), a set bit in TPR0<10:0> specifies a test. Only one test at a time can be specified. Test results are passed through TPR2 and TPR3 as described below. After a test is completed, TPR0 is cleared. The user application should be reloaded after any of these tests are performed.

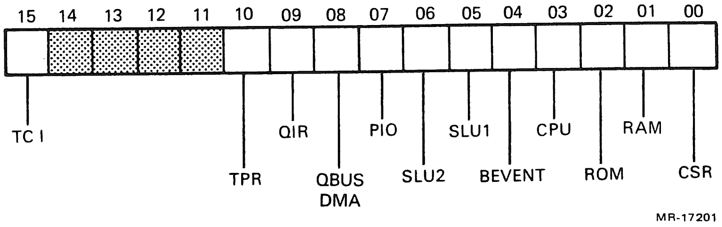


Figure 3-4   TPR0 as a Test Register

Bits	Name	Description
15	TC I	Test/Control indicator - When set, TPR0 is used as a test register. When cleared, TPR0 is used as a control register.
14:11		Not used (read as zeros)
10	TPR	TPR test - When set, performs read and write tests on the local side of the TPR file. TPR4 through TPR15 are set at zero upon completion of this test.
9	QIR	QIR test - When set, tests the Q-Bus interrupt mechanism. The address of the interrupt vector must be in TPR3 before this test is run.
8	DMA	DMA controller test - When set, tests the on-board DMA controller by performing DMA transfers between memory locations.
7	PIO	PIO test - When set, tests the parallel I/O port and its associated timers. A loopback connector must be installed on J4 before this test is run.

Bits	Name	Description
6	SLU2	SLU2 test - When set, tests the multiprotocol serial controller. Loopback connectors for J1 and J2 must be installed before this test is run. A one in TPR3 indicates that SLU2 channel A will be tested. A two in TPR3 indicates that SLU2 channel B will be tested. A zero in TPR3 indicates that SLU2 channels A and B will be tested.
5	SLU1	SLU1 test - When set, tests the console serial line. A loopback connector must be installed on J3 before this test is run.
4	BEVENT	BEVENT test - When set, verifies that the line clock interrupt (BEVENT) can be enabled, asserted, and disabled. The interrupt associated with BEVENT is at priority level 6 with a vector of 100.
3	CPU	CPU test - When set, tests the on-board J-11 microprocessor.
2	ROM	ROM test - When set, performs a checksum test of the on-board ROM. TPR3 must be set to 1, if the user (P)ROM and the native firmware are to be tested, or set at zero, if only the native firmware is to be tested.
1	RAM	RAM test - When set, performs a non-destructive test of all on-board RAM.
0	CSR	CSR test - When set, performs read tests on KXJCSRA through KXJCSRJ, and the control/status registers for all the other on-board I/O devices.

TPR2<10:0> indicate which test(s), if any, have failed (see Figure 3-5). A set bit indicates a failed test. Note the correspondence between TPR2<10:0> and the tests specified by TPR0<10:0> respectively. TPR2<15:11> are unused.

TPR2 and TPR3 provide detailed information about certain failed tests, as summarized in Table 3-1. Bits <15:12> of TPR3 (Figure 3-6) contain an octal value (see Figure 3-5), which indicates the last test if it failed (Figure 3-6).

This value equals the TPR2 error bit that was set plus one. For example, if the RAM test failed, TPR3 bits <15:12> equal two, since TPR2 bit 1 is the RAM test error bit.

Table 3-1 TPR2 and TPR3 as Test Result Registers

Failed Test	TPR2 Bit Set	TPR3 Bit Set	Definition
CSR	0	0	Bus error at CSR address
		1	Bus error at QIR address
		2	Bus error at TPR address
		3	Bus error at SLU1 address
		4	Bus error at SLU2 address
		5	Bus error at SLU2 counter/timer address
		6	Bus error at PIO address
		7	Bus error at DMA controller address
		8-11	Undefined
RAM	1	0	RAM error. TPR1 will have the NXM or parity error flags set if appropriate. Otherwise the data read did not match the data written.
		1-11	Undefined
ROM	2	0	Native firmware checksum error
		1	User application (P)ROM checksum error
		2	No user (P)ROM space exists
		3-11	Undefined
CPU	3	0-11	Undefined
CLOCK	4	0	Undefined
		1	Clock interrupt not masked at level 6
		2	Clock doesn't interrupt
		3	Can't shut it off
		4-11	Undefined
SLU1	5	0	Undefined
		1	XMTR interrupt not masked at level 4
		2	XMTR interrupt not received
		3	RCVR interrupt not masked at level 4
		4	RCVR interrupt not received
		5	Recieved data incorrect
		6	No RCVR done, loopback open
		7-11	Undefined
SLU2	6	0	Undefined
		1	SLU2 counter/timer 2 doesn't interrupt
		2	Asych mode, data transfer incomplete
		3	Synch mode, EOF-SDLC not received
		4	Synch mode, data transfer incomplete
		5	Synch/asynch mode, received data incorrect
		6-7	Undefined
		8	Status incorrect or no interrupt request with "request to send" set
		9	Status incorrect or no interrupt request with "RS-422" set
		10	Status incorrect or no interrupt request with "terminal in service" set

Table 3-1   TPR2 and TPR3 as Test Result Registers (Cont)

Failed Test	TPR2 Bit Set	TPR3 Bit Set	Definition
PIO	7	0	Undefined
		1	Reset state incorrect
		2	Timer did not start
		3	Timer never stops
		4	Interrupt not masked at level 4
		5	Interrupt not received
		6	Loop timeout, data transfer incomplete
		7	Received data incorrect
		8-11	Undefined
DMA	8	0	Undefined
		1	Q-Bus address undefined, access not tested
		2	Channel interrupt not received
		3	DMA channel hung (TC/EOP both cleared)
		4	DMA aborted (EOP = 1 = NXM)
		5	DMA data error
		6-11	Undefined
QIR	9	0	Q-Bus vector not defined
		1	Q-Bus interrupt request never posted
		2	Undefined
		3	Local interrupt not masked at level 5
		4	Local interrupt not received
		5	Interrupt acknowledge didn't clear interrupt request
		6-11	Undefined
TPR	10	0	The data read did not match the data written
		1	NXM error
		2-11	Undefined

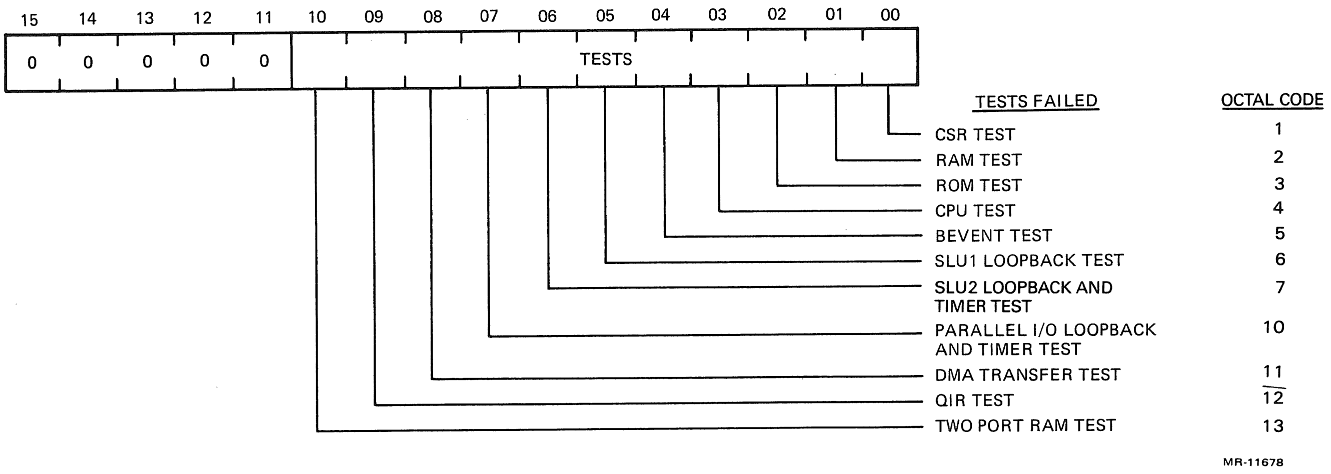


Figure 3-5   TPR2 as a Test Result Register

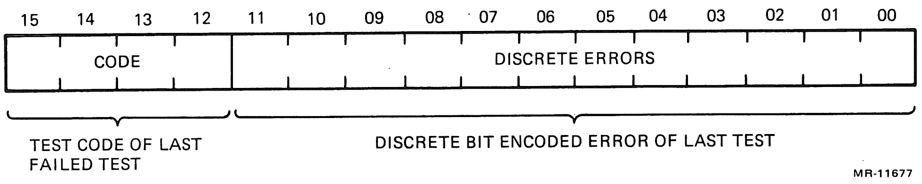

15
14
13
12
11
10
09
08
07
06
05
04
03
02
01
00
CODE
DISCRETE ERRORS
TEST CODE OF LAST FAILED TEST
DISCRETE BIT ENCODED ERROR OF LAST TEST
MR-11677

Figure 3-6   TPR3 as a Test Result Register

3.2.3.2.3 TPR0 as a Q-Bus ODT Register -- When the KXJ11-CA is in Q-Bus ODT mode, TPR0 is interpreted as shown in Figure 3-7. This interpretation of TPR0 continues until bit 15 is set.

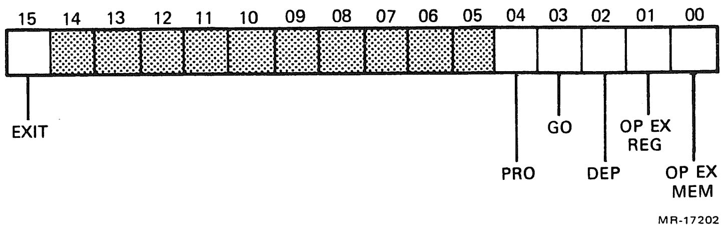


Figure 3-7   TPR0 as a Q-Bus ODT Register

Bits	Name	Description
15	EXIT	Exit ODT - When set, Q-Bus ODT mode is exited. The KXJ11-CA then awaits a command from the arbiter.
14:5		Not used (read/write)

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th></tr>
<tr><td>4</td><td>PRO</td><td>Proceed - When set, the context of an interrupted program is restored and the execution of the program resumes at the address specified by the restored PC.</td></tr>
<tr><td>3</td><td>GO</td><td>Start program - When set, a restart operation is performed and the execution of the program begins at the PC address passed through TPR3. The system bus is initialized. A RESET instruction is executed to initialize the local I/O devices. The MMR0<15:13,0> and MMR3 registers are set at zero by Micro/ODT, and are set at zero on this system when RESET is executed. The following registers are cleared: PS, PIRQ, CPUERR, Memory System Error, and Floating Point Status. The system Memory Error Register (177744) is cleared by Micro/ODT but does not exist on the KXJ11-CA.</td></tr>
<tr><td>2</td><td>DEP</td><td>Deposit - When set, the contents of TPR2 are loaded into the current open memory location or register.</td></tr>
<tr><td>1</td><td>OP EX REG</td><td>Open and examine register - When set, the register specified by TPR3 is opened and its contents are loaded into TPR2. The registers are encoded in TPR3 as follows (note that bit 11 of the PS selects the register set):
<table>
<tr><th>Code</th><th>Register</th></tr>
<tr><td>000000</td><td>R0 or R0'</td></tr>
<tr><td>000001</td><td>R1 or R1'</td></tr>
<tr><td>000002</td><td>R2 or R2'</td></tr>
<tr><td>000003</td><td>R3 or R3'</td></tr>
<tr><td>000004</td><td>R4 or R4'</td></tr>
<tr><td>000005</td><td>R5 or R5'</td></tr>
<tr><td>000006</td><td>R6, SP</td></tr>
<tr><td>000007</td><td>R7, PC</td></tr>
<tr><td>000010</td><td>PSW</td></tr>
</table>
Any other code will set bit 15 of TPR1, indicating a command error.</td></tr>
<tr><td>0</td><td>OP EX MEM</td><td>Open and examine memory - When set, opens a memory location and deposits its contents in TPR2. The address of the memory location has 22 bits. The six most significant bits are obtained from the six least significant bits of TPR2. The lower 16 bits are obtained from TPR3.</td></tr>
</table>

3.2.3.3 TPR1 -- The firmware uses TPR1 to record KXJ11-CA errors (see Figure 3-8). This register is read-only from the Q-Bus but can be read or written by the on-board J-11.

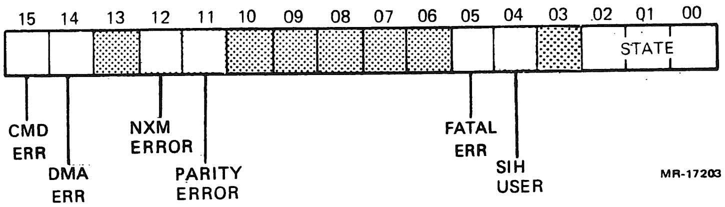


Figure 3-8 TPR1

Bits	Name	Description
15	CMD ERR	Command error - Set when an error is detected during the execution of a command.
14	DMA ERR	DMA error - Set when the DTC aborts after a DMA LOAD command from TPR0 (bit 1) has been issued.
13		Not used (read/write)
12	NXM ERR	Non-existent memory error - Set when a non-existent memory trap occurs while Q-Bus ODT or self-tests are running.
11	PARITY ERR	Parity error - Set when parity trap occurs while Q-Bus ODT or self-tests are running.
10:6		Not used (read/write)
5	FATAL ERR	Fatal error - Set when a fatal error is detected during auto self-test, or Q-Bus controlled self-test. The KXJ11-CA becomes unavailable and does not respond to any commands from the arbiter, except the setting of TPR0 bit 14, which causes a hardware reset.
4	SIH USER	Special interrupt handling - Set when user application code handles special interrupts. Clear when firmware handles special interrupts.
3		Not used (read/write)

Bits	Name	Description
2:0	STATE	State - Reflects the state of the KXJ11-CA: 000 Zero State - KXJ11-CA not available. No commands should be sent from the Q-Bus. 001 Power-up Auto Self-test - The KXJ11-CA is performing its auto self-tests. 010 Dedicated Test State - The boot/self-test switch is set to either 7 or 15. No commands should be sent from the Q-Bus. 011 Q-Bus ODT Mode - The KXJ11-CA is participating in a Q-Bus ODT operation. Only Q-Bus ODT commands should be sent from the Q-Bus. 100 Waiting For Command - The KXJ11-CA is idle and waiting for a command from the arbiter. 101 Loading Application From TU58 - The KXJ11-CA is loading (or attempting to load) a boot block from the TU58 connected to the console serial line. 110 Reserved - This state is reserved for future use by Digital Equipment Corporation. 111 Executing User Application Code - The KXJ11-CA is executing a user application program. Note that STATE is indeterminate when J-11 console ODT is active.

3.2.3.4 TPR2 -- TPR2 is used to pass parameters required to execute commands. See the description of TPR0 (Section 3.2.3.2) for the commands and parameters that involve TPR2. This register can be read or written by both the Q-Bus and the on-board J-11.

3.2.3.5 TPR3 -- TPR3 is used to pass parameters required to execute commands or perform tests. Refer to Sections 3.2.3.2 through 3.2.3.2.3 for the commands, tests, and parameters that involve TPR3. Upon hardware reset, TPR3 has the following format (Figure 3-9).

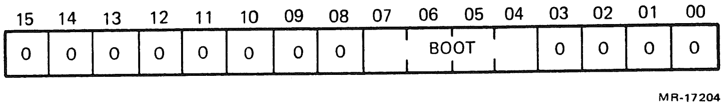


Figure 3-9 TPR3 Format During Hardware Reset

Bits	Name	Description
15:8		Not used (read as zeros)
7:4	BOOT	Boot/self-test switch - Reflects the encoded switch position of the boot/self-test switch.
3:0		Not used (read as zeros)

This register can be read or written by both the Q-Bus and the on-board J-11.

3.2.3.6 TPR4 Through TPR15 -- If the TPRs are enabled and the appropriate enable bit in KXJCSRD is set, registers TPR4 through TPR15 are used by the user's application to pass status and control information between the Q-Bus arbiter and the KXJ11-CA. All the TPRs may be read or written by the on-board J-11. From the Q-Bus, however, TPR1, TPR5, and TPR9 are read-only and the other TPRs are read/write.

Writes to TPR4, TPR8, and TPR12 from the Q-Bus cause maskable level 5 interrupts. The vectors associated with these interrupts are 120, 124, and 134 respectively. The status of the enables and interrupt requests are contained in the KXJCSRD register.

3.2.4 PROM and Firmware Control

The operation of the KXJ11-CA is controlled by firmware that resides in two 8K X 8 PROMs (Intel 2764 or equivalent). The firmware occupies 8 KB of PROM space. The other 8 KB of PROM space is available for the user's application program. Note that PROM data at addresses 2140000 through 2177777 also appears at other addresses. The address space from 2140000 - 2177777 is duplicated three times for the 8K X 8 PROMs (see Figure 3-10).

If you want to enter your application in PROM, you will need a PROM programmer and a program called DECPROM or its equivalent. Using these items, blast your new PROMs that contain both a copy of the firmware and the application. The procedure for doing this is explained in Appendix C.

The KXJ11-CA can also accommodate two 16K X 8 (Intel 27128 or equivalent) or two 32K X 8 (Intel 27256 or equivalent) PROMs if you need more than 8 KB of PROM for your application. Figure 3-11 shows how address space is allocated for the 16K X 8 PROMs. The data at addresses 2100000 through 2177777 is duplicated once. Figure 3-12 shows the address space allocation for the 32K X 8 PROMs. There is no duplication of address space when using the 32K x 8 PROMs. The firmware always occupies 8 KB of space.

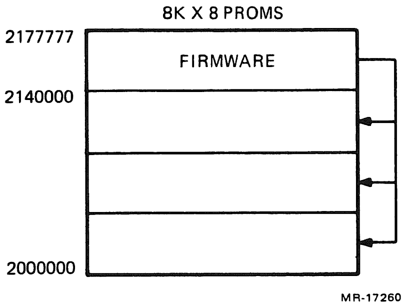


Figure 3-10 PROM Space Allocation - 8K x 8 PROMs

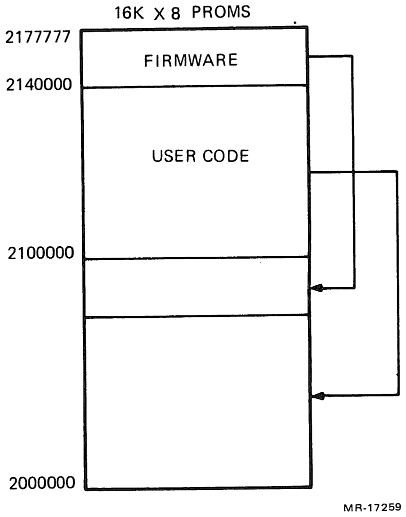


Figure 3-11 PROM Space Allocation - 16K x 8 PROMs

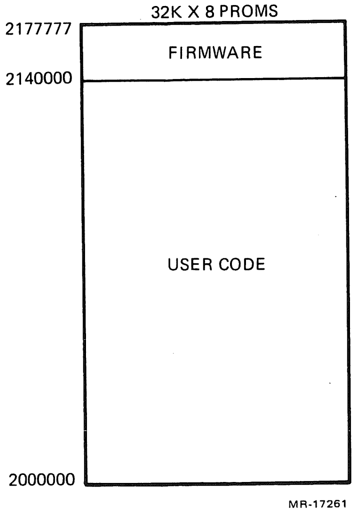


Figure 3-12 PROM Space Allocation - 32K x 8 PROMs

3.2.4.1 Firmware Usage Considerations -- During power-up, the firmware assigns PAR7 and PDR 7 in kernel I (instruction) space to map to the I/O page. Do not modify the mapping registers in your program if you continue to use your native firmware. In addition, if you enable kernel D (data) space mapping, then PAR7 and PDR 7 in kernel data space must also be mapped to the I/O page.

If firmware commands are going to be issued while a user application is running on the KXJ11-CA, there must always be 128 bytes available on the kernel stack. Upon entering the firmware through a special interrupt condition (see Section 3.5), 128 bytes should be used on the kernel stack to save the user's context and for internal work space.

Do not alter MMR0, MMR3, or any of the kernel mapping registers using QBUS ODT. The registers may examined, but a command error will be returned in TPR1 if an attempt is made to write to any of them.

When using firmware to load or start a user application program, no code or data should be loaded into locations 157600 through 157777. This is where the firmware stack resides.

In order for the arbiter to communicate with the native firmware, bit 6 of KXJCSRJ (NMI enable) must be set. Note that if the KXJ11-CA executes a RESET instruction, KXJCSRJ is cleared. Bit 6 of KXJCSRD (TPR enable) must also be set. When the firmware is waiting to receive a command or is in the process of exiting to user code, it sets bit 6 of KXJCSRJ and bit 6 of KXJCSRD.

3.2.4.1.1 Self-tests -- The native firmware determines whether or not self-tests are to be performed, depending upon the setting of the boot/self-test switch (see Section 2.2.1). A subset of the self-tests is called the auto self-tests, and includes tests of the on-board CPU, RAM, ROM, control/status registers (CSRs), two-port RAM (TPR) registers, the line clock (BEVENT) mechanism, and the local DMA mechanism. Test results are deposited in TPR2 and TPR3 at the completion of the auto self-tests (Section 3.2.3.2.2). Test status is displayed on the LEDs (Table 2-9).

3.2.4.1.2 Booting -- If the self-tests run successfully, the firmware is then ready to boot the user's application specified by the setting of the boot/self-test switch (Section 2.2.1). The application can be booted either from ROM or from a TU58 tape drive. If the application is booted from ROM, the native firmware transfers control to the application in ROM by emulating a trap to location 24. If the application is booted from a TU58, a boot block is transferred to the KXJ11-CA via SLU1 (the console serial port). If the first byte of the received boot block matches an octal value of 240 to 277, the boot block is considered valid and execution is transferred to location 0. If a match does not occur, the boot operation is repeated alternately on TU58 units 0 and 1.

An application may also be loaded into RAM through either shared memory, or by requesting the firmware to perform a DMA load operation that transfers the application from external Q-Bus memory to the KXJ11-CA local memory. Once the application is in local memory, KXJ11-CA processor execution is transferred to the application by requesting the firmware to perform a Q-Bus ODT "Go" command.

3.2.4.2 Native Firmware Vs. User-Designed Firmware -- The KXJ11-CA is shipped with firmware that is referred to as "native firmware". Native firmware provides the KXJ11-CA with the functions described in the sections that follow. The handling of Q-Bus exceptions, interrupts, and resets are all functions which involve the native firmware.

If you wish to design your own firmware, your user-designed firmware should have an entry point at physical location 173004. An entry point in firmware for location 173000 should also be provided for power-up handling.

3.2.5 CPU ID Switch

A hex encoded ID switch is used to select either standalone or IOP mode of operation. ID numbers range from 0 through 15, with 0 and 1 signifying standalone operation, and 2 through 15 signifying system usage of the Q-Bus. The ID switch code can be read via KXJCSRC. There are two jumpers (M3-M4 and M5-M6) that correspond with the address width (16-, 18-, or 22-bit) of the Q-Bus backplane being used. These jumpers determine the size of the memory decode required for shared memory.

3.2.6 DMA Controller

A 16-bit DMA controller is addressed by the local processor as an I/O device. The DMA controller has two independent channels and can perform transfers between any local 22-bit address and any 16-, 18-, or 22-bit Q-Bus bus address. Transfers can also be performed between any two local 22-bit addresses or any two Q-Bus addresses. Word, high byte, and low byte operations are supported locally. Only word operations are supported across the Q-Bus interface. Either the source or the destination may have incrementing, decrementing, or fixed addresses. Words may be compared with a mask register as they flow through the DMA controller or as they are read. DMA operations can be interleaved with the local processor and the other channel, or may occur in various burst sizes. Channel 0 or channel 1 can service hardware requests from SLU2 or the PIO, or can be invoked by software commands after certain mask control bits are cleared.

3.2.7 Wake-Up Circuit

The wake-up circuit provides automatic generation of the INIT signal on the J-11 to initialize the LOCAL system (i.e., the KXJ11-CA board only). The wake-up circuit does not support power down sequencing, and assumes that +5V and +12V rise together. Q-Bus signals BDCOK and BPOK are used to synchronize the Q-Bus with the LOCAL system bus. With Vcc rising in approximately 30 - 40 ms, power-up occurs in approximately 400 ms for standalone mode, and 550 ms for peripheral processor mode.

3.2.8 KXJ11-CA Control and Status Registers

The KXJ11-CA has eight registers that are used to monitor and control the overall operation of the board. They are the KXJ11 Control and Status Registers, and both are described in the sections that follow. All the KXJ11 Control/Status Registers are contained in the on-board gate arrays.

3.2.8.1 KXJ11 Control/Status Register A (KXJCSRA) -- Control/Status Register A (Figure 3-13) is used to monitor and control SLU2 and the real time clock (RTC). This register is cleared upon hardware reset.

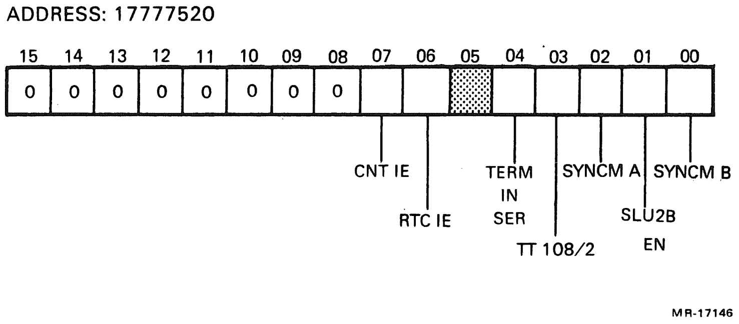


Figure 3-13 KXJ11 Control/Status Register A

Bits	Name	Description
15:8		Not used (read as ones)
7	CNT IE	Programmable counter interrupt enable - When set, interrupts from programmable timer/counter 2 are enabled. When cleared, these interrupts are inhibited.
6	RTC IE	Real time clock interrupt enable - When set, interrupts from the on-board line-time clock (LTC) are enabled. When cleared, these interrupts are disabled.
5		Not used (read/write)
4	TERM IN SER	Terminal in service - For use with modems. When set, Terminal In Service (IS) is asserted and incoming calls can be connected. When cleared, IS is not asserted.
3	TT108/2	Modem connected - For use with modems. When set, Terminal Ready (TR) is asserted. When cleared, TR is not asserted.

Bits	Name	Description
2	SYNCM A	Clock select channel A - When set, SLU2 channel A receives its clock from the on-board baud rate generator. When cleared, channel A receives its clock from an external source.
1	SLU2BR EN	Party line enable - Used when the KXJ11-CA is configured for party line operation. When set, SLU2 channel B cannot receive party line data. When cleared, party line data reception for channel B is enabled.
0	SYNCM B	Clock select channel B - When set, SLU2 channel B receives its clock from the on-board baud rate generator. When cleared, channel B receives its clock from an external source.

3.2.8.2 KXJ11 Control/Status Register B (KXJCSRB) -- Control/Status Register B (Figure 3-14) is used to monitor the state of the boot/self-test switch, the base address jumper, the bus size jumpers, and the SLU2 modem test function. The register is read-only, with the exception of bits <7:4>, which are read/write.

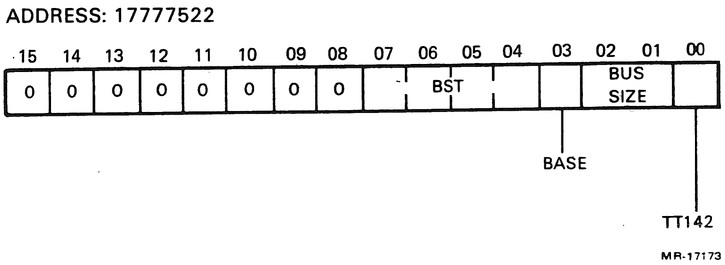


Figure 3-14 KXJ11 Control/Status Register B

Bits	Name	Description
15:8		Not used (read as zeros)
7:4	BST	Boot/self-test switch - Contains the encoded value of the boot/self-test switch position (see Section 2.2.1 for a description of the boot/self-test switch). 0000 corresponds to switch position 0, 0001 corresponds to switch position 1, and so on. These bits are read/write. BST is loaded with the encoded value of the boot/self-test switch during hardware reset, and can be changed by the user's software. Exercise caution when writing BST, since it controls the mapping of ROM to high or low addresses.
3	BASE	Base address jumper - When set, indicates that the Q-Bus base address jumper is installed (see Section 2.2.3). This bit is loaded during hardware reset and cannot be changed by software.
2:1	BUS SIZE	Bus size jumpers - Indicates the Q-Bus size jumper settings (see Section 2.2.2). This bit is loaded upon hardware reset and cannot be changed by software.

Bus Size	Address Bits Used
00	22
01	16
10	18
11	Reserved

0	TT142	Modem test - When set, indicates that the modem connected to SLU2 channel A is in test mode. When cleared, indicates that the modem is not in test mode. Cleared upon hardware reset.

3.2.8.3 KXJ11 Control/Status Register C (KXJCSRC) -- KXJ11 Control/Status Register C (Figure 3-15) contains information on the state of the CPU ID switch and the state of the on-board LEDs.

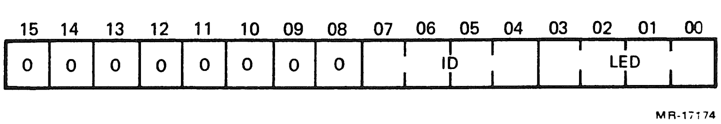


Figure 3-15 KXJ11 Control/Status Register C

Bits	Name	Description
15:8		Not used (read as zeros)
7:4	ID	CPU ID switch - Contains the encoded value of the CPU ID switch position. 0000 corresponds to switch position 0, 0001 corresponds to switch position 1, and so on. These bits are read-only. These bits are loaded during hardware reset and cannot be changed by software.
3:0	LED	LED state - Each bit determines the state of one of the four on-board LEDs. LEDs 4 through 1 correspond to bits <3:0> respectively. If a bit is set, the LED is ON. If a bit is cleared, the LED is OFF. These bits are read to determine the state of the LEDs, or are written to set the LEDs. These bits are set to one during hardware reset.

3.2.8.4 **KXJ11 Control/Status Register D (KXJCSRD)** -- KXJ11 Control/Status Register D (Figure 3-16) monitors and controls the QIR, the TPRs, and the Q-Bus reset/interrupt mechanism. This register is cleared during reset. Access is read/write.

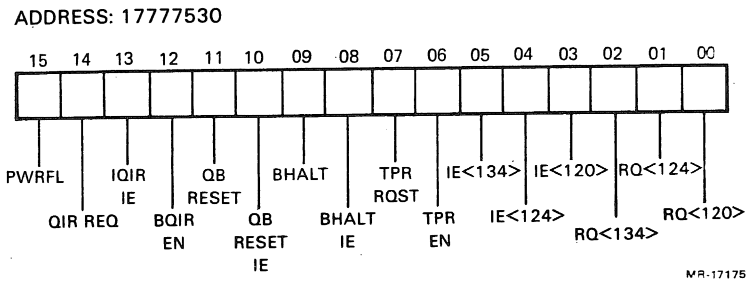


Figure 3-16 KXJ11 Control/Status Register D

Bits	Name	Description
15	PWRFL	Power fail - When set, the Q-Bus has deasserted BPOK, indicating a power failure. When clear, indicates that BPOK is asserted. This bit should be cleared after a PWRFL special interrupt.

Bits	Name	Description
14	QIR REQ	QIR request - When set, indicates that the QIR has been written and that a Q-Bus interrupt is pending. Clearing QIR REQ after it has been set clears the pending request to the Q-Bus. The deassertion of the Q-Bus signal BIAKI clears QIR REQ. This bit cannot be set by the user. QIR REQ has no meaning if the KXJ11-CA is operating in standalone mode.
13	IQIR IE	QIR interrupt enable for J-11 - When set, the on-board J-11 receives a level 5 interrupt request for vector 130 when any of the following situations occur. 1. When BIAKI is asserted as part of the Q-Bus interrupt handling sequence. 2. When bit 14 (QIR REQ) is set and then cleared before the Q-Bus interrupt has been serviced. 3. When BINIT is asserted before the Q-Bus interrupt has been serviced. If IQIR IE is set, the arbiter causes a local level 5 interrupt when it acknowledges a QIR interrupt.
12	BQIR EN	QIR interrupt enable for Q-Bus master - When set, enables the Q-Bus master to participate in Q-Bus interrupt handling. When BQIR EN is set and J-11 writes the vector, the Q-Bus master receives a level 4 interrupt request for the vector in the QIR register. When cleared, Q-Bus interrupt requests are blocked from reaching the Q-Bus master.
11	QB RESET	Q-Bus reset - Set when bit 10 (QB RESET is IE) and bit 6 KXJCSRJ (NMI EN) is set, a special interrupt condition is that is handled by the KXJ11-CA native firmware (see Section 3.5 for details). This bit should be cleared after a QB RESET special interrupt.
10	QB RESET IE	Q-Bus reset interrupt enable - When set, enables the KXJ11-CA to detect the assertion of the Q-Bus signal BINIT. When set, and when BINIT is asserted, bit 11 (QB RESET) is set. When cleared, the KXJ11-CA is prevented from responding to the assertion of BINIT.

Bits	Name	Description
9	BHALT	Bus halt - Set when bit 8 (BHALT IE) and bit 6 KXJCSRJ (NMI EN) is set, and BHALT is asserted. When BHALT is set, a special interrupt condition exists that is handled by the KXJ11-CA native firmware (see Section 3.5). Firmware handles this by emulating a trap through vector 24. This bit should be cleared after a BHALT special interrupt.
8	BHALT IE	Bus halt interrupt enable - When set, enables the KXJ11-CA to detect the assertion of the Q-Bus signal BHALT. When set, and when BHALT is asserted, bit 9 (BHALT) is set. When cleared, the KXJ11-CA is prevented from responding to the assertion of BHALT.
7	TPR RQST	TPR restart request - Set by a Q-Bus write to TPR0 when KXJCSRJ bit 6 (NMI EN) is set. This indicates an exception condition that is handled by the KXJ11-CA native firmware. This bit cannot be set directly. Once TPR RQST is set, subsequent writes to TPR0 do not cause more exceptions. This bit is cleared by the KXJ11-CA native firmware when the command has completed execution.
6	TPR EN	TPR enable - When set, allows the contents of the TPR file to be accessed from the Q-Bus. When cleared, forces the Q-Bus to read zeros from the TPR file (writes will time out). All TPRs except TPR0 are enabled and disabled by this bit. TPR0 is always accessible from the Q-Bus.
5:3	IE<134> IE<124> IE<120>	TPR interrupt enables - When each bit is set, a TPR interrupt enables a level 5 interrupt request to occur when a particular TPR is written. IE<134> controls interrupt requests from TPR12 for vector 134. IE<124> controls interrupt requests from TPR8 for vector 124. IE<120> controls interrupt requests from TPR4 for vector 120. When a bit is cleared, the corresponding interrupt request is blocked.

Bits Name Description

2:0 RQ<134>
RQ<124>
RQ<120> TPR request flags - When each bit is set, TPR request flags indicate that a particular TPR has been written. RQ<134>, RQ<124>, and RQ<120> corresponds to TPR12, TPR8, and TPR4 respectively. If the corresponding IE<134>, IE<124>, or IE<120> bit is also set, a level 5 interrupt occurs when the TPR is written.

3.2.8.5 KXJ11 Control/Status Register E (KXJCSRE) -- Control/status register E is a dummy register provided for software compatibility with the corresponding reserved register on the KXT11-CA. This register can be read and written, but writes to the register do not affect KXJ11-CA operation, and reads to the register always produce zeros. The address of KXJCSRE is 17777526.

3.2.8.6 KXJ11 Control/Status Register F (KXJCSRF) -- KXJ11 Control/Status Register F (Figure 3-17) defines the lower limit of the shared memory space accessible to the Q-Bus. The upper limit is defined by KXJ11 Control/Status Register H (see Section 3.2.8.7). This register is initialized to a value of 177600 upon power-up.

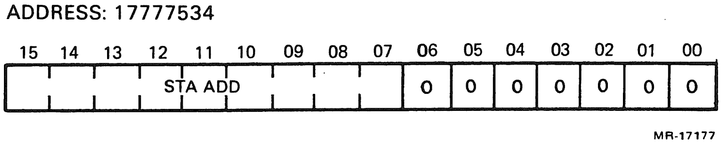


Figure 3-17 KXJ11 Control/Status Register F

Bits Name Description

15:7 STA ADD Starting address - Contains the most significant nine bits of a Q-Bus starting address. The starting address defines the beginning of the shared memory space on this board that is accessible to the Q-Bus. STA ADD corresponds to BDAL<21:13> at address time. These bits are read/write and are unaffected by a hardware reset.

6:0 Not used (read as zeros)

3.2.8.7 KXJ11 Control/Status Register H (KXJCSRH) -- KXJ11 Control/Status Register H (Figure 3-18) defines the upper limit of the shared memory space accessible to the Q-Bus. The register also contains the number of blocks in this memory space. The lower limit is defined by KXJ11 Control/Status Register F (see Section 3.2.8.6). This register is initialized to a value of 177777 during power-up.

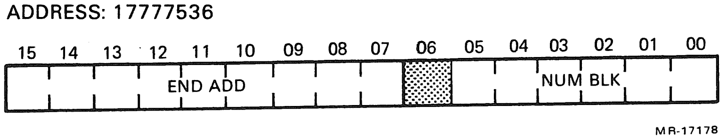


Figure 3-18 KXJ11 Control/Status Register H

Bits	Name	Description
15:7	END ADD	Ending address - Contains the nine most significant bits of a Q-Bus ending address for the board's shared memory. The ending address is defined as the first address of the last 8K block of Q-Bus accessible shared memory (see Section 3.8.1 for an example of the use of END ADD). The KXJ11-CA compares END ADD with addresses on the Q-Bus, to determine the Q-Bus addresses that refer to the board's shared memory. END ADD corresponds to BDAL<21:13> at address time. These bits are read/write and are unaffected by a hardware reset.
6		Not used (read/write)
5:0	NUM BLK	Number of blocks - Contains a value that represents the number of 8 KB blocks in the shared memory space that are accessible to the Q-Bus. This value is derived from the starting address of the Q-Bus shared memory and the number of blocks to be shared (see Section 3.8). Since the shared memory is 512 KB, up to 64 8 KB blocks can be specified. These bits are read/write and are unaffected by a hardware reset.

3.2.8.8 KXJ11 Control/Status Register J (KXJCSRJ) -- KXJ11 Control/Status Register J (KXJCSRJ) (Figure 3-19) enables and disables the non-maskable interrupts (power fail, BINIT, BHALT, and TPR0 writes). KXJCSRJ also indicates if timeouts for DMA or bus-locked operations have occurred. KXJCSRJ determines if the baud rate for SLU1 is under software control, and determines if shared memory can be accessed from the Q-Bus. KXJCSRJ also specifies parity characteristics for the on-board RAM. This register is read/write and is cleared upon hardware or software reset.

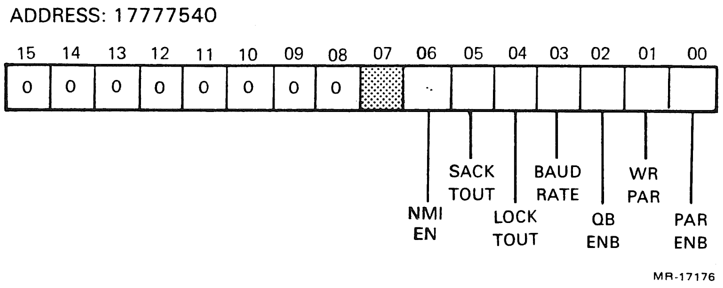


Figure 3-19 KXJ11 Control/Status Register J

Bits	Name	Description
15:8		Not used (read as zeros)
7		Not used (read/write)
6	NMI EN	Non-maskable interrupt enable - When written with a 1, enables recognition of interrupts from the following sources: power failures, the assertion of BINIT or BHALT, and interrupts that result from writing TPR0. When NMI EN is written with a 0, disables recognition of the interrupts from the sources listed previously.
5	SACK TOUT	SACK timeout - Set if a DMA request to the Q-Bus is not granted in the alotted time (approximately 140 us). May occur when the Q-Bus is heavily loaded with DMA activity from multiple devices. Writing a 1 has no effect on this bit. This bit must be explicitly cleared by writing a zero.
4	LOCK TOUT	Lock timeout - Set when a bus locked instruction (WRTLCK, TSTSET, or ASRB) is executing locally and access to the Q-Bus cannot be obtained in the alotted time (approximately 140 us). Writing a 1 has no effect on this bit. This bit must be explicitly cleared by writing a zero.

Bits	Name	Description
3	BAUD RATE	Baud rate - When set, the baud rate for SLU1 is under software control, according to the value written to PB in the Console Transmitter Status Register (XCSR). When cleared, the baud rate is determined by the SLU1 baud rate jumpers.
2	QB ENB	Q-Bus enable - When set, enables the Q-Bus to access the KXJ11-CA shared memory that it has been allocated. When cleared, prevents Q-Bus access to the shared memory.
1	WR PAR	Write parity - When set, generates wrong parity on writes to the on-board RAM. Does not write wrong parity on writes from MicroODT.
0	PAR ENB	Parity enable - When set, enables parity errors to be detected. If a parity error is detected, a non-maskable parity interrupt occurs with an associated vector of 114. When cleared, parity errors are ignored.

3.2.9 Q-Bus Interrupt Register (QIR)

The Q-Bus Interrupt Register (QIR) (Figure 3-20) is used by the KXJ11-CA to interrupt the arbiter. When the KXJ11-CA initiates a Q-Bus interrupt, it loads an interrupt vector into the Q-Bus Interrupt Register. This causes bit 14 in KXJ11-CA Control/Status Register D (see Section 3.2.8.4) to be set. It then asserts BIRQ4 on the Q-Bus, if bit 12 (BQIR EN) of Control/Status Register D is set. The KXJ11-CA drives the content of the QIR register on the Q-Bus, if it receives BIAKI, and bit 12 of Control/Status Register D is set. The receipt of BIAKI clears bit 14 in KXJCSRD. This register is write-only.

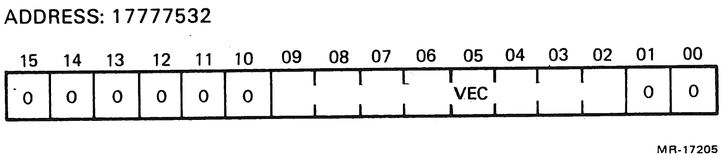


Figure 3-20 Q-Bus Interrupt Register (QIR)

Bits	Name	Description
15:10		Must be zero
9:2	VEC	Vector - Contains the interrupt vector used to service the KXJ11-CA's interrupt to the Q-Bus. These bits are not affected by a hardware reset.
1:0		Must be zero

3.2.10 Maintenance Register

The Maintenance Register (Figure 3-21) indicates which halt and power-up options were selected by the user. It also indicates the status of the Q-Bus signal BPOK. This register is read-only and is unaffected by a hardware reset.

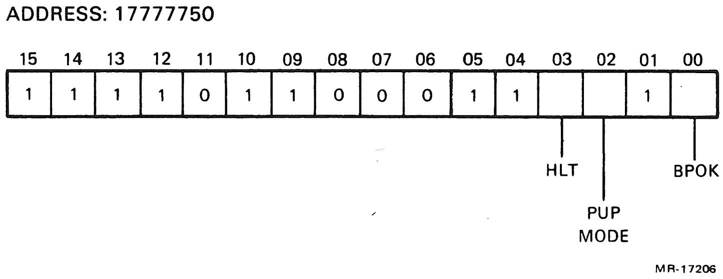


Figure 3-21 Maintenance Register

Bits	Name	Description
15:8		Is 11110110
7:4		Is 0011, indicating the CPU code for the KXJ11-CA.
3	HLT	Halt - When set, indicates that M15 and M14 are not jumpered together. When cleared, indicates that M15 and M14 are jumpered together. The M15 - M14 jumper determines what action the KXJ11-CA will take if a HALT instruction is executed in kernel mode (see Section 2.2.7).
2	PUP MODE	Power-up mode - When set, indicates that M16 and M17 are not jumpered together. When cleared, indicates that M16 and M17 are jumpered together. The M16 - M17 jumper determines how the KXJ11-CA will act when the board is powered up or reset (see Section 2.2.8).
1		Must be 1
0	BPOK	BPOK status - Set when the Q-Bus signal BPOK is asserted.

3.2.11 Program Interrupt Request (PIRQ) Register

The Program Interrupt Request (PIRQ) Register provides seven levels of software interrupt capability for the on board J-11 microprocessor. An interrupt is queued by setting one of bits <15:9>, which correspond to interrupt priority levels 7 through 1 respectively. Bits <7:5> and <3:1> are set by the on board J-11 to the encoded value of the highest pending request. When the interrupt request is granted, the J-11 traps through location 240 in kernel I space. The user's interrupt service routine must clear the appropriate PIRQ bit before exiting. The format of the PIRQ is shown in Figure 3-22.

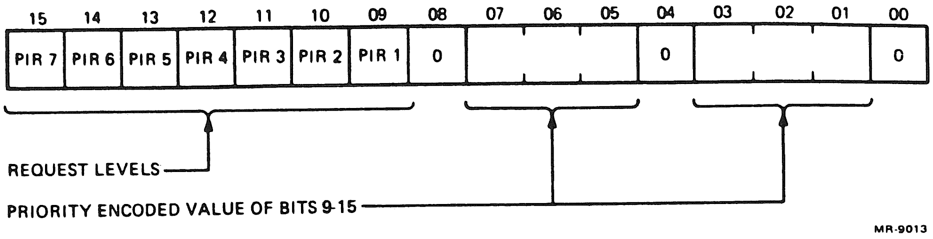


Figure 3-22 PIRQ Register

Bits <15:9> can be read or written. Bits <7:5> and <3:1> are read-only. The other bits are read as zeros.

3.2.12 CPU Error Register

The CPU Error Register (Figure 3-23) identifies the source of a trap through location 4. Refer to the DCJ11 Microprocessor User's Guide (EK-DCJ11-UG) for details on handling the traps. This register is read/write.

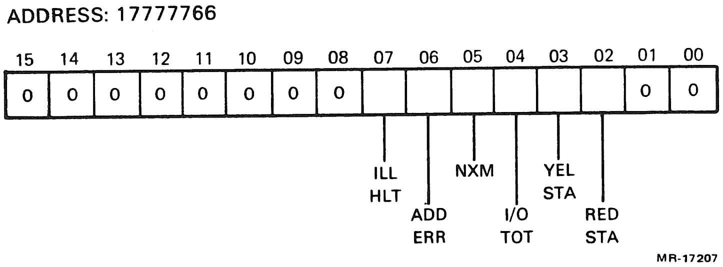


Figure 3-23 CPU Error Register

Bits	Name	Description
15:8		Not used (read as zeros)
7	ILL HLT	Illegal halt - Set when execution of a HALT instruction is attempted in user or supervisor mode, or attempted in kernel mode when M14 and M15 are jumpered together.
6	ADD ERR	Address error - Set when a word access is made to an odd byte address, or when an instruction fetch from a J-11 internal register is attempted.
5	NXM	Non-existent memory - Set when reference is made to a non-existent memory address.
4	I/O TOT	I/O bus timeout - Set when reference is made to a non-existent I/O page address.
3	YEL STA	Yellow stack violation - Set when a yellow zone stack overflow occurs.
2	RED STA	Red stack trap - Set when a red stack trap occurs.
1:0		Unused (read as zeros)

3.2.13 Processor Status Word (PSW)

The Processor Status Word (Figure 3-24) contains the current and previous operational modes, the J-11 general-purpose register set being used, the current priority level, condition codes, and the trace trap bit. All bits in this register are read/write except bits <10:9> (which are read-only but are not used).

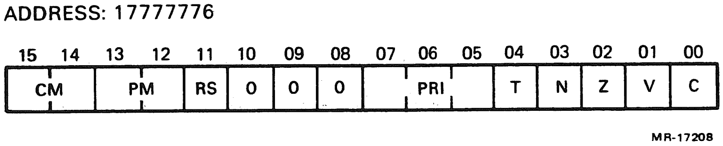


Figure 3-24 Processor Status Word (PSW)

Bits	Name	Description
15:14	CM	Current mode - Displays the current operational mode: **CM** **Mode** 00 Kernel 01 Supervisor 10 Illegal 11 User
13:12	PM	Previous mode - Displays the previous operational mode using the same codes that are used for CM.
11	RS	Register set - When set, set 1 (R0' through R5') of the J-11 general-purpose registers is used. When cleared, set 0 (R0 through R5) is used.
10:8		Not used (read as zeros)
7:5	PRI	Priority - Determines the hardware interrupt priority level. **PRI** **Priority Level** 111 7 110 6 101 5 100 4 011 3 010 2 001 1 000 0
4	T	Trace trap - When set, causes a trap to location 14 at the end of the current instruction. When cleared, disables the trace trap function.
3	N	N bit - Set if the result of the previous instruction was negative.
2	Z	Z bit - Set if the result of the previous instruction was zero.
1	V	V bit - Set if the previous instruction resulted in an arithmetic overflow.
0	C	C bit - Set if the previous instruction resulted in a carry of its most significant bit.

3.2.14 Console Asynchronous Serial I/O

The console asynchronous serial line interface (based on the DLART chip) provides program or jumper selectable baud rates (300 to 38.4K baud), real-time clock outputs (800 Hz, 60 Hz, and 50 Hz), eight data bits, no parity, one stop bit, and break detection that causes the J-11 to enter ODT if the BREAK enable jumper (M12-M13) is installed. A break detect bit appears in the RBUF of a UART during the time a BREAK condition exists on the line. Also featured are RS422/RS423/RS232-C EIA interfaces, a 10-pin interface connector, and an optional EIA to 20ma conversion kit utilizing the DLV11-KA. There is no reader run pulse generation or 110 baud rate input.

3.2.15 Synchronous/Asynchronous Serial I/O

A two-channel multiprotocol serial communications controller (uPD7201) supports asynchronous, character-oriented synchronous, and bit-oriented synchronous protocols, programmable character size, parity, CRC generation and checking, BREAK detect, framing error detection, automatic detection and generation of SYNC characters, auto hunt, and external or internal programmable baud rates from 110 to 76.8K baud. The primary channel (SLU2 channel A) is provided with type SR (send-receive) RS449(CCITT) electrical interface and modem control lines. The secondary channel (SLU2 channel B) is a synchronous/asynchronous secondary channel with type DT (data and timing only) RS449(CCITT) electrical interface. In addition, this second channel can be operated in a 16 node party-line configuration.

3.2.16 Parallel I/O

Twenty programmable parallel I/O lines are provided, with programmable direction control of IEEE-488 electrical standard compliant input buffers, and either passive pull-up drivers or TTL compatible drivers. There are three parallel I/O ports: two 8-bit data ports and one 4-bit control port. Features include three interrupt requests and handshake control for either polled, interrupt conditional control, three-wire, or bidirectional operation. Three programmable 16-bit timers are provided with either internal control and interrupt, or external buffered control lines.

3.2.17 -12V Charge Pump

Local serial I/O drivers and receivers require a negative 12V bias. This is provided by an on board charge pump operating at 614.4 KHz. The charge pump is zener diode regulated.

3.3 Q-BUS INTERFACE

The Q-Bus interface can be considered from two perspectives: from the perspective of the Q-Bus and from the perspective of the local KXJ11-CA bus. There are two distinct portions of the KXJ11-CA that may be partitioned in any way the user sees fit. The I/O page addressable TPR file's address is determined by the base address and the CPU ID switch setting, while the addresses in shared memory can be defined by the user. From the Q-Bus, the KXJ11-CA looks like a 16 word I/O page addressable register file that may be logically partitioned into a transmit status and command section, and a receive command and status section. The file provides a control communication path. The Q-Bus master can read and write from/to any of three defined command registers and has read status from any of three associated read only status registers. All other file words may be written or read. TPR4, TPR8, and TPR12, when written from the Q-Bus, can either flag the J-11 through KXJCSRD or interrupt the J-11 with unique level five interrupts. In addition, when the first file word is written, a nonmaskable J-11 restart trap is generated. This is a priority command channel.

The Q-Bus sends messages or asks for status through the TPR file. The J-11 can respond either by directly reading or writing the file, or by invoking the data path controller (DTC) to move data across the bus interface and signal the arbiter when completed. The local processor can signal the Q-Bus arbiter at will by writing into its Q-Bus interrupt register (QIR), to generate a level 4 interrupt to the Q-Bus arbiter, or by writing into the two port register for polled operations.

The KXJ11-CA DMA controller can address any portion of the Q-Bus 22-bit address space. Shared memory is visible from the local bus as a contigious physical memory with an address range of 00000000 - 01777777 (512 KB). All 512 KB can be shared. The addressing from the Q-Bus is determined by the host and stored in two internal registers, KXJCSRF and KXJCSRH. These registers contain the starting address (KXJCSRF), a value for the number of blocks, and the ending address (KXJCSRH). Shared memory is enabled and disabled by KXJCSRJ. The locally mapped Q-Bus memory may be allocated in 8 KB contigious increments in any non I/O address range. The range selected must also be on 8 KB boundaries, which allow a user a total of sixty-four 8 KB pages. The shared memory area is located at the top of local RAM.

3.4 KXJ11-CA INTERRUPTS

There are three general categories of interrupts which involve the KXJ11-CA: interrupts from the Q-Bus to the KXJ11-CA, interrupts from the KXJ11-CA to the Q-Bus, and local interrupts by on-board KXJ11-CA I/O devices. This section describes the KXJ11-CA's role in each type of interrupt. Special interrupt handling by the firmware is discussed in Section 3.5.

3.4.1 Interrupts From The Q-Bus To The KXJ11-CA

A Q-Bus device can interrupt the KXJ11-CA by writing TPR4, TRP8, or TPR12. If the TPRs and the TPR interrupts are enabled (as determined by bits 6:3 of KXJCSRD), a write to any of these three registers from the Q-Bus causes a level 5 interrupt. The vectors associated with the interrupts are located at logical addresses 120, 124, and 134 respectively in kernel I space. The sequence of events during a Q-Bus interrupt is illustrated in Table 3-2.

Table 3-2 Interrupts from the Q-Bus to the KXJ11-CA

Q-Bus Device or Arbiter	KXJ11-CA
Writes TPR4, TPR8, or TPR12	If TPRs and TPR interrupts are enabled, handles a level 5 interrupt with the following vector. TPR Vector TPR4 120 TPR8 124 TPR12 134

3.4.2 Interrupts From The KXJ11-CA To The Q-Bus

The KXJ11-CA can interrupt devices on the Q-Bus (including the arbiter) through a register in the on-board GAP gate array (DC7037B) called the QIR or Q-Bus Interrupt Register. The format of the QIR is described in Section 3.2.9.

In order for the KXJ11-CA to interrupt the Q-Bus through the QIR, bit 12 of KXJCSRD (BQIREN) must first be set. If this bit is cleared, interrupts from the KXJ11-CA cannot be posted.

If KXJCSRD bit 12 is set, a write to the QIR sets KXJCSRD bit 14 (QIR REQ), which causes the Q-Bus signal BIRQ4 to be asserted and generates a level 4 interrupt on the Q-Bus. At some later time, a Q-Bus device (or arbiter) asserts the signal BIAKI to acknowledge the interrupt, and reads the contents of the QIR for the appropriate vector.

The assertion of BIAKI (acknowledgement of interrupt) clears KXJCSRD bit 14. If KXJCSRD bit 13 is set, the assertion of BIAKI posts a level 5 local interrupt request with a vector of 130. The user must ensure that there is a vector at 130 that points to a routine that will handle the interrupt. The routine handles the interrupt and the operation is complete. Table 3-3 summarizes the sequence of events.

Table 3-3 Interrupts from the KXJ11-CA to the Q-Bus

Q-Bus Device or Arbiter	KXJ11-CA
	Writes QIR with vector when KXJCSRD<12> is set
	Sets KXJCSRD<14>
	Asserts BIRQ4
Asserts BIAKI	Reads QIR
	Handles a local level 3 interrupt with a vector of 130

3.4.3 Local Interrupts From On-Board Devices

The KXJ11-CA on-board devices that can post local interrupts to the on-board J-11 include the data transfer controller (DTC), the parallel I/O port (PIO), the console serial line (SLU1), the multiprotocol serial controller (SLU2), the SLU2 counter/timer, the real-time clock (RTC), TPR4, TPR8, TPR12, and the QIR.

Interrupts from local devices are all handled in the same general way.

1. The local device posts an interrupt to the on-board J-11 through the J-11's IRQ lines,

2. The J-11 performs an interrupt acknowledge cycle and reads a vector that points to an interrupt service routine,

3. The routine handles the interrupt, and

4. Operation resumes.

The DTC and the PIO share a common interrupt request line. The DTC has the higher priority of the two devices (these two devices are daisy-chained), and allows the PIO to acknowledge an interrupt only if there are no DTC interrupts pending.

Table 3-4 is a summary of all the interrupts handled locally by the KXJ11-CA, their relative priorities, and the vectors associated with each interrupt. Within a priority level, the interrupt with highest priority is listed first.

The vectors for DTC interrupts and PIO interrupts shown in Table 3-4 are the defaults set by the native firmware. These vectors are programmable.

Table 3-4 Summary of KXJ11-CA Local Interrupts

Priority	Vector(s)	Interrupt Type
Programmable	240	PIRQ
6	100	Real-Time Clock
6	104	SLU2 Counter/Timer 2 (watchdog timer
5	120	Interrupt from Q-Bus
5	124	Interrupt from Q-Bus
5	130	Interrupt to Q-Bus (via QIR)
5	134	Interrupt from Q-Bus
4	224,230*	DTC Interrupt
4	200,204,210*	PIO/PIO Timer Interrupt
4	60	Console (SLU1) Receiver
4	64	Console (SLU1) Transmitter
4	70	MPSC (SLU2) Communication

*Default values. May be changed by the user.

3.5 SPECIAL INTERRUPT HANDLING

The KXJ11-CA native firmware is designed to handle four types of special interrupts. A special interrupt occurs when the enable bit (KXJCSRJ bit 6) is set and

1. A command is issued (TPR0 is written), or

2. A power fail occurs (BPOK is deasserted), or

3. A Q-Bus halt occurs (BHALT is asserted) and the corresponding enable bit (KXJCSRD bit 8) is set, or

4. A Q-Bus initialization is performed (BINIT is asserted) and the corresponding enable bit (KXJCSRD bit 10) is set.

During an special interrupt, the KXJ11-CA switches to kernel mode, pushes the current PC and PS on the kernel stack, forces the PC to 173004, forces the PS to 340, clears bit 6 of KXJCSRJ (NMI EN), and begins to execute code. Typically, the KXJ11-CA firmware handles the special interrupt. If the user has his own code for special interrupt handling, he should make sure that the entry point for this code is at the kernel logical location 173004. The firmware handles a special interrupt as follows:

1. TPR1 bit 4 selects either user code or firmware to handle the special interrupt. If TPR1 bit 4 = 1, control is passed to user code through locations 24 and 26, and user code handles the special interrupt. If TPR1 bit 4 = 0, control is retained by the KXJ11-CA firmware that handles the special interrupt. The steps that follow assume the KXJ11-CA firmware handles the special interrupt.

2. The cause of the special interrupt is determined by the contents of KXJCSRD.

3. If the special interrupt is caused by a power failure, the firmware traps through locations 24 and 26 and allows user code to handle the interrupt.

4. If the special interrupt is caused by the assertion of BHALT or BINIT and the appropriate enable bit is set, the firmware traps through locations 24 and 26 and allows user code to handle the interrupt.

5. If the special interrupt is caused by issuing a command, the firmware determines which command has been issued by looking at TPR0 and then executes the command. The arbiter may need to load command parameters into TPR2 and TPR3 before it issues the command.

6. After the special interrupt is handled, the firmware sets the enable bit (bit 6 of KXJCSRJ) to enable another special interrupt. If the user's code is processing the special interrupt, it is important that the proper exit sequence be used upon completion. First, the event indicator (BHALT, BINIT, or PWRFL) must be cleared in KXJCSRD. Then, the enable bit must be set in KXJCSRJ. Finally, an RTI instruction must be executed.

3.6 KXJ11-CA RESETS

There are two ways in which the KXJ11-CA can be reset or reinitialized.

1. By executing a RESET instruction. This is called a software reset.

2. By the assertion of the local power-up signal PUP. This is called a hardware reset.

The sections that follow explain the causes and effects of the two types of resets.

3.6.1 Software Reset

When a RESET instruction is executed by the on-board J-11, the various components of the KXJ11-CA are affected as summarized in Table 3-5.

Table 3-5 KXJ11-CA Software Reset

Component	Effect
SLU1 (DLART)	The DLART input INIT is asserted, which clears interrupt enables and clears bits 2 and 0 of the SLU's XCSR. Refer to the DLART Data Sheet for details of DLART behavior when INIT is asserted.
PIO	The PIO inputs ZDS and ZAS are asserted which resets the PIO. Refer to the PIO Technical Manual for details of · PIO behavior when DS and AS are asserted.
DTC	The DTC inputs DS and AS are asserted, which resets the DTC. Refer to the DTC Technical Manual for details of DTC behavior when DS and AS are asserted.
J-11	Memory management is disabled and the J-11 executes a RESET instruction. Refer to the DCJ11 User's Guide for details of J-11 behavior when RESET is executed.
SLU2 (MPSC)	Unaffected.
SLU2 Timers	Unaffected.
RTC	Interrupts disabled (KXJCSRA<6> cleared)
KXJCSRA KXJCSRB KXJCSRC	Unaffected.
KXJCSRD	Cleared.
KXJCSRE KXJCSRF KXJCSRH	Unaffected.
KXJCSRJ	Cleared. This disables shared memory and special interrupts. It has other effects as described in Section 3.2.8.8.
Boot/Self-test Switch	No effect on software resets.
LEDs	Unaffected.
ID Switch	No effect on software resets.
Pending Interrupts	Cleared.

3.6.2 Hardware Reset

Hardware resets are caused by the assertion of the local power-up signal PUP. A hardware reset occurs when any of the following conditions occur.

1. If the KXJ11-CA is in standalone mode (as determined by the setting of the ID switch), a hardware reset occurs when an on-board wake-up circuit detects the presence of +5V DC power.

2. If the KXJ11-CA is not in standalone mode (as determined by the setting of the ID switch), a hardware reset occurs when the on-board wake-up circuit detects the assertion of the Q-Bus signal BDCOK. A hardware reset is also caused by writing a 1 to TPR0 bit 14 from the Q-Bus.

During a hardware reset, the various components of the KXJ11-CA are affected as summarized in Table 3-6.

Table 3-6 KXJ11-CA Hardware Reset

Component	Effect
SLU1 (DLART)	The DLART input TEST is asserted, which resets the DLART. Refer to the DLART Data Sheet for details of DLART behavior when TEST is asserted.
PIO	The PIO inputs ZDS and ZAS are asserted, which resets the PIO. Refer to the PIO Technical Manual for details of PIO behavior when ZDS and ZAS are asserted.
DTC	The DTC inputs ZDS and ZAS are asserted which resets the DTC. Refer to the DTC Technical Manual for details of DTC behavior when ZDS and ZAS are asserted.
J-11	The J-11 input INIT is asserted. Refer to the DCJ11 Microprocessor User's Guide (EK-DCJ11-UG) for details of J-11 behavior when INIT is asserted. Jumper M17-M16 determines if control is passed to the firmware (location 173000) or to ODT after power-up is complete.
SLU2 (MPSC)	The MPSC input RESET is asserted. Refer to the MPSC Data Sheet for details of MPSC behavior when RESET is asserted.
SLU2 Timers	Initialize themselves upon power-up.
RTC	Interrupts disabled (KXJCSRA<6> cleared)

Table 3-6 KXJ11-CA Hardware Reset (Cont)

Component	Effect
KXJCSRA KXJCSRB KXJCSRC KXJCSRD KXJCSRE KXJCSRJ	Initialized to power-up values. See specific register descriptions for details. All writeable bits are cleared.
KXJCSRF	Initialized to a value of 177600 upon power-up, and are unaffected by a hard reset from the Q-Bus.
KXJCSRH	Initialized to a value of 177777 upon power-up, and are unaffected by a hard reset from the Q-Bus.
Boot/Self-test Switch	If firmware is executed upon power-up, this switch specifies the function performed.
LEDs	All on
ID Switch	Value is loaded into KXJCSRC, thus affecting the Q-Bus address.
Pending Interrupts	Cleared.

3.7 MEMORY MANAGEMENT ARCHITECTURE

NOTE

This section was written for readers familiar with PDP-11 memory management concepts. For further details on memory management, refer to Chapter 4 of the DCJ11 Microprocessor User's Guide (EK-DCJ11-UG-PRE).

The KXJ11-CA implements the full PDP-11 memory management and protection architecture with extensions for extended direct addressing. The KXJ11-CA memory management registers include Page Address Registers (PARs), Page Descriptor Registers (PDRs), and Memory Management Registers 0 through 3 (MMR0 - MMR3). MMR0 through MMR3 are contained in the on-board J-11 microprocessor. The PARs and PDRs are located in physical memory. These registers are described in the sections that follow.

3.7.1 Page Address Registers (PARs)

There is a total of 48 PARs, with eight PAR's alloted for each of the following: kernel I space, supervisor I space, user I space, kernel D space, supervisor D space, and user D space. Each PAR contains a page address field (PAF) that specifies the starting address of a page as a block number in physical memory.

> NOTE
>
> Kernel I space and D space PAR7 is mapped to the I/O page by the firmware. This mapping must not be altered.

The format of a PAR is shown in Figure 3-25.

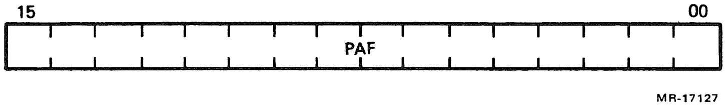


Figure 3-25 Page Address Register (PAR)

3.7.2 Page Descriptor Registers (PDRs)

There is a total of 48 PDRs, with eight PAR's alloted for each of the following: kernel I space, supervisor I space, user I space, kernel D space, supervisor D space, and user D space. Each PDR contains information on expansion direction, page length, and access control. The format of a PDR is shown in Figure 3-26.

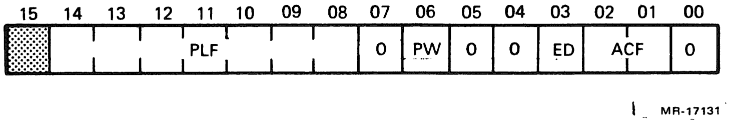


Figure 3-26 Page Descriptor Register (PDR)

Bits	Name	Description
15		Not used (read/write)
14:8	PLF	Page length field - Specifies the block number, which defines the page boundary (see bit 3). The block number of the virtual address is compared with PLF to detect length errors. An error occurs when expanding upwards if the block number is greater than PLF, and when expanding downwards if the block number is less than PLF.

Bits	Name	Description
7		Not used (read as zero)
6	PW	Page written - When set, this page has been modified since it was loaded into memory. This bit is cleared when the PAR or PDR of this page is written.
5:4		Not used (read as zeros)
3	ED	Expansion direction - When set, this page expands downwards from block number 127 to include blocks with lower addresses. When cleared, this page expands upwards from block number 0 to include blocks with higher addresses.
2:1	ACF	Access control field - Contains the access code for this page. **ACF** **Access** 00 Non-resident - abort all accesses 01 Read only - abort on write attempt 10 Not used - abort all accesses 11 Read/write access
0		Not used (read as zero)

3.7.3 Memory Management Register 0 (MMRO)

MMRO contains status and control information for the memory management unit. This register is read-only. Figure 3-27 illustrates the format of MMRO.

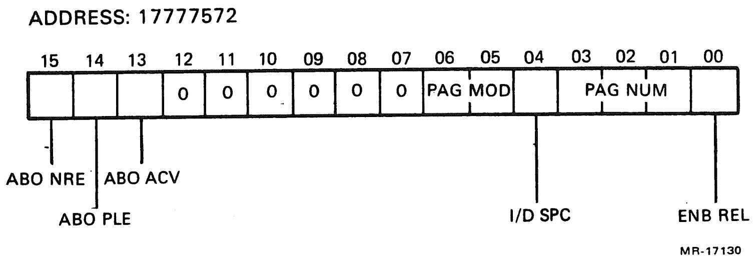


Figure 3-27 Memory Management Register 0 (MMRO)

Bits	Name	Description
15	ABO NRE	Abort non-resident - Set when an access attempt is made to a page with an access control field key of 0 or 2. Also set by attempting to use memory relocation with a current processor mode (PS<15:14>) of 2 (illegal). ABO NRE is set when PAG MOD equals 2.
14	ABO PLE	Abort page length - Set when an access attempt is made to a page with a block number outside the range specified by the page's PDR. Also set by attempting to use memory relocation with a current processor mode (PS<15:14>) of 2 (illegal).
13	ABO ACV	Abort access violation - Set when attempting to write a read-only page (that is, the access control field equals 1.)
12:7		Not used (read as zeros)
6:5	PAG MOD	Page mode - Indicates the CPU mode associated with the page causing an abort. **PAG MOD** **Mode** 00 Kernel 01 Supervisor 10 Illegal Mode 11 User If an illegal mode is specified, ABO NRE is set.
4	I/D SPC	Page address space - When set, a D space mapping operation was attempted when an abort occurred. When cleared, an I space mapping operation was attempted when an abort occurred.
3:1	PAG NUM	Page number - Contains the page number of a reference causing a memory management abort.
0	ENB REL	Enable relocation - When set, memory management is enabled and address relocation occurs. When cleared, memory management is disabled and addresses are neither relocated nor protected. Cleared by RESET instruction.

3.7.4 Memory Management Register 1 (MMR1)

MMR1 records the autoincrementing or autodecrementing of any general purpose register (GPR) during an instruction, including references through the program counter (PC). This register is cleared at the beginning of an instruction. Whenever a GPR is autoincremented or autodecremented, the register number and amount (in 2's complement notation) by which the register was modified is written into MMR1. The low byte of MMR1 is written first. The format of MMR1 is shown in Figure 3-28.

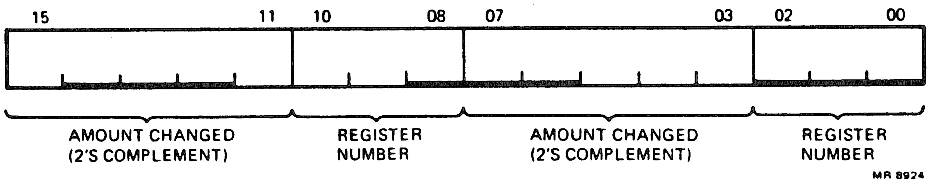

15
11
10
08
07
03
02
00
AMOUNT CHANGED (2'S COMPLEMENT)
REGISTER NUMBER
AMOUNT CHANGED (2'S COMPLEMENT)
REGISTER NUMBER
MR 8924

Figure 3-28 Memory Management Register 1 (MMR1)

3.7.5 Memory Management Register 2 (MMR2)

MMR2 is also called the virtual program counter (VPC) and is loaded with a 16-bit virtual address at the beginning of each instruction fetch. This register is read-only.

3.7.6 Memory Management Register 3 (MMR3)

MMR3 (Figure 3-29) enables and disables data space mapping for kernel, user, and supervisor modes. It also controls I/O mapping, 18-bit/22-bit mapping, and whether requests for Call to Supervisor Mode instruction are enabled. This register is read/write and is cleared during a hardware reset.

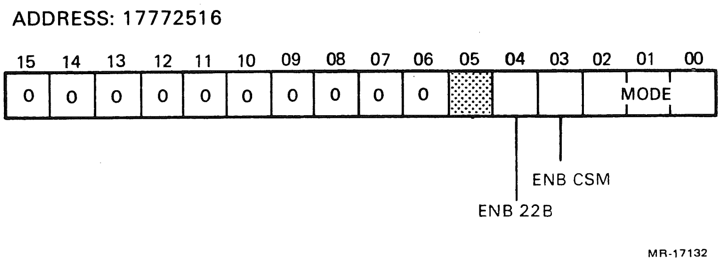


Figure 3-29 Memory Management Register 3 (MMR3)

Bits	Name	Description
15:6		Not used (read as zeros)
5		Not used (read/write)
4	ENB 22B	Enable 22-bit mapping - When this bit is set and memory management is enabled (i.e., bit 0 of MMR0 is set), 22-bit mapping is used. When this bit is cleared and when memory management is enabled, 18-bit mapping is used. This bit has no effect when memory management is disabled.
3	ENB CSM	Enable Call to Supervisor Mode - When this bit is set, a Call to Supervisor (CSM) instruction may be executed. When this bit is cleared, the execution of a CSM instruction causes a trap through location 10 in kernel I space.
2:0	MODE	Mode bits - enable and disable kernel, supervisor, and user D space as shown: **Bit** **Meaning** MODE<2> = 0 Disable kernel D space MODE<2> = 1 Enable kernel D space MODE<1> = 0 Disable supervisor D space MODE<1> = 1 Enable supervisor D space MODE<0> = 0 Disable user D space MODE<0> = 1 Enable user D space

3.8 SHARED MEMORY

The KXJ11-CA contains 512 KB of on-board RAM. The RAM can be configured as "shared memory" that can be accessed by devices on the Q-Bus as well as the on-board J-11 microprocessor. Shared memory could be used, for example, in an application where the arbiter needs to access RAM that is read or written locally.

Shared memory can be configured under software control by loading KXJ11-CA Control/Status Registers KXJCSRF and KXJCSRH (Sections 3.2.8.6 and 3.2.8.7), and enabled by setting KXJCSRJ<2>. KXJCSRF and KXJCSRH contain values that specify the starting address, ending address, and number of blocks for the shared memory area. This section explains how to derive the values you need to load into KXJCSRF and KXJCSRH once you have determined which Q-Bus addresses are to be associated with shared memory. The section that follows explains the mechanics of how the registers are loaded by the arbiter or by the on-board J-11.

When configuring shared memory, make sure that there are no overlapping Q-Bus addresses. That is, Q-Bus addresses must be unique.

3.8.1 Shared Memory Organization

Shared memory consists of one or more 8 KB blocks of RAM. Since there is 512 KB of RAM on the KXJ11-CA, the maximum number of shared memory blocks is 64. Each block must start on an 8 KB boundary. On the KXJ11-CA, the last block is the highest 8 KB of RAM (1777777 - 1760000), the next to last block is the next highest 8 KB (1757777 - 1740000), and so on. All blocks of shared memory are contiguous. The shared memory space is located at the top of local RAM. The local starting address is (2000000 - N*20000) octal, where N is the number of blocks.

The algorithms for determining the contents of KXJCSRF and KXJCSRH are as follows.

```
     CSRF:          (Q-Bus starting address/100) octal

     CSRH<15:7>:    [(Ending address) AND 17720000]/20000
     CSRH<5:0>:     [Negate[(Q-Bus starting address)/20000 +
                    (number of blocks)]] AND 77 octal
```

The sections that follow illustrate how these values are determined and used.

3.8.2 Defining One Block of Shared Memory

Suppose you want to define one 8 KB block of addresses as shared memory. The following example illustrates how this would be done.

In this example, you want to define one 8 KB block of shared memory starting at Q-Bus address 100000. In this case (Figure 3-30), addresses 100000 through 117777 on the Q-Bus correspond to KXJ11-CA shared memory addresses 1760000 through 1777777. What values do we need to load into KXJCSRF and KXJSCRH?

The value for the "starting address" that we need to load into KXJCSRF is obtained by shifting the starting Q-Bus address to the right six bits. Figure 3-31 shows the relationship between KXJCSRF and the 22-bit Q-Bus address. Plugging in our Q-Bus address of 100000 (octal), we see that 1000 (octal) should be loaded into KXJCSRF. Bits 6:0 of KXJCSRF are not used and read as zeros.

The value for the "ending address" that we need to load into KXJCSRH bits 15:7 (END ADD) is obtained by taking the first (Q-Bus) address in the last block and shifting it to the right 13 bits. In this case, we are working with only one block, so the first block is the only block. The first address in the block shifted to the right 13 bits yields 000000100 for bits 15:7.

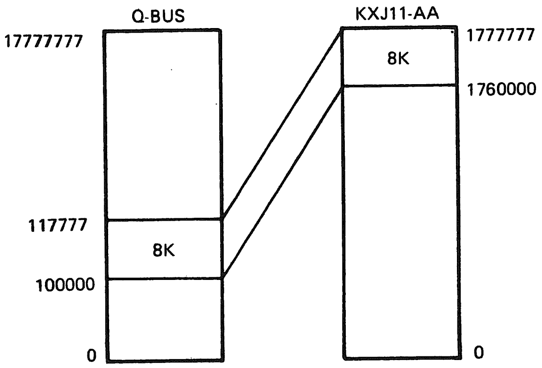


Figure 3-30 Defining One Block of Shared Memory

	15	14	13	12	11	10	9	8	7	6:0
STARTING Q-BUS ADDRESS	21	20	19	18	17	16	15	14	13	X

MR-17255

Figure 3-31 Control Register Bits/Q-Bus Address Relationship

The value for the "number of blocks" that we need to load into KXJCSRH bits 5:0 is obtained by extracting bits 18:13 of the Q-Bus starting address, adding the number of blocks, and two's complementing (negating) the result.

Q-Bus address 100000 bits 18:13	000100
Add number of blocks	+ 1

	000101
Negate	111011

The value 111011 is loaded into bits 5:0 of KXJCSRH. The KXJ11-CA interprets this value as one block.

3.8.3 Defining Two Blocks of Shared Memory

Suppose you want to define two blocks of shared memory. Assume that the range of Q-Bus addresses assigned to shared memory in this case is 100000 through 137777. On the KXJ11-CA, the corresponding two blocks are contiguous in RAM and reside at addresses 1740000 through 1777776. The relationship between the Q-Bus addresses and the KXJ11-CA addresses is illustrated in Figure 3-32.

Find the correct values to load into KXJCSRF and KXJCSRH.

The value for the "starting address" that we need to load into KXJCSRF is obtained by shifting the starting Q-Bus address to the right six bits. This yields 1000 (octal) for KXJCSRF. Bits 6:0 of KXJCSRF are not used and read as zeros.

The value for the "ending address" that we need to load into KXJCSRH bits 15:7 (END ADD) is obtained by taking the first (Q-Bus) address in the last block and shifting it to the right 13 bits. In this case, the first address of the last block is 120000. Shifting the address to the right 13 bits yields 000000101 for bits 15:7.

The value for the "number of blocks" that we need to load into KXJCSRH bits 5:0 is obtained by extracting bits 18:13 of the Q-Bus starting address, adding the number of blocks, and negating the result.

```
Q-Bus address 100000 bits 18:13        000100
Add number of blocks                   +   10
                                       ------
                                       000110

Negate                                 111010
```

The value 111010 is loaded into bits 5:0 of KXJCSRH. The KXJ11-CA interprets this value as two blocks.

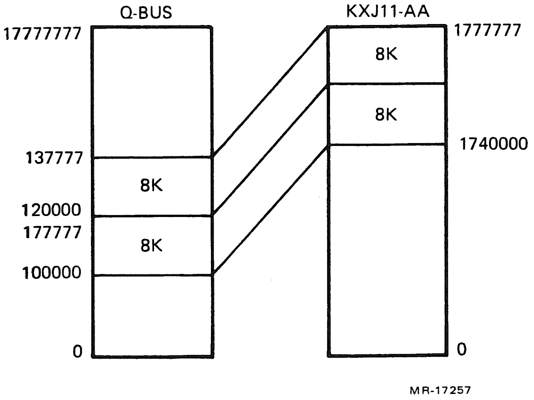


Figure 3-32 Defining Two Blocks of Shared Memory

3.8.4 Defining 64 Blocks of Shared Memory

Suppose you want to define all 64 blocks of RAM as shared memory. Assume that the range of Q-Bus addresses assigned to shared memory in this case is 1000000 through 2777777. On the KXJ11-CA, the corresponding blocks are contiguous in RAM and reside at addresses 0 through 1777777. The relationship between the Q-Bus addresses and the KXJ11-CA addresses is illustrated in Figure 3-33.

The correct values to load into KXJCSRF and KXJCSRH are

KXJCSRF = 10000 (octal)

Since the first (Q-Bus) address of the last block is 2760000,

KXJCSRH bits 15:7 = 001011111

KXJCSRH bits 5:0 are determined as follows.

```
     Q-Bus address 1000000 bits 18:13        100000
     Add number of blocks                 + 1000000
                                            -------
                                            1100000

     Negate                                  100000
```

Carries are ignored.

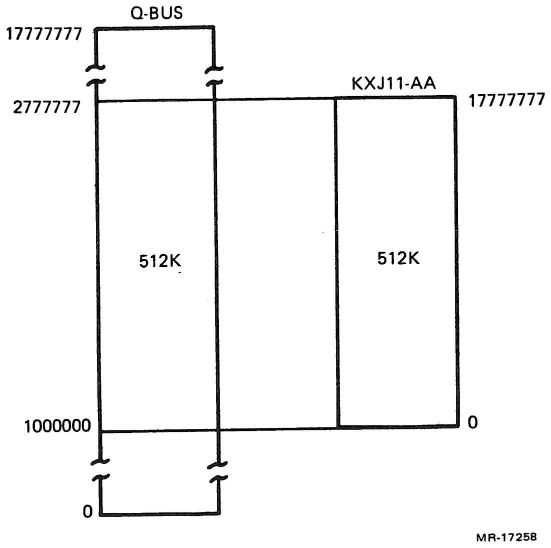


Figure 3-33 Defining 64 Blocks of Shared Memory

3.8.5 Enabling and Disabling Shared Memory

Shared memory can be enabled and disabled by either the on-board J-11 or by an arbiter command. When shared memory is enabled, the relationship between Q-Bus addresses and KXJ11-CA RAM addresses is defined by the values in KXJCSRF and KXJCSRH. This section describes how the on-board J-11 and arbiter can enable and disable shared memory.

Once KXJCSRH and KXJCSRF are set up, the on-board J-11 enables and disables shared memory simply by writing bit 2 of KXJCSRJ. When this bit is set, shared memory is enabled. When the bit is cleared, shared memory is disabled. The shared memory configuration values might be known at startup time, or they can be passed from the arbiter to the KXJ11-CA through one of the TPR user communication channels.

When the arbiter wants to enable or disable shared memory through the firmware, the process is somewhat more involved. To enable shared memory, the following events occur.

1. The arbiter determines that the KXJ11-CA is ready to receive a command. This occurs when TPR1<2:0> = 100, and TPR0<15:0> = 0. This is sometimes called the "waiting for command" state.

2. The arbiter writes bits 21:13 of the Q-Bus starting address into TPR2<8:0> and writes zeros into TPR2<15:9>.

3. The arbiter writes the number of blocks of shared memory minus one into TPR3<5:0>. For example, TPR3<5:0> = 000000 for one block of shared memory, TPR3<5:0> = 000001 for two blocks of shared memory, and so on. The arbiter writes zeros into TPR<15:6>.

4. The arbiter sets TPR0 bit 6. Only bit 6 should be set. If the arbiter were to set more than one bit at a time in TPR0, an error would result (and would be recorded in TPR1). Setting bit 6 causes the KXJ11-CA firmware to configure and enable shared memory. The data in TPR2 and TPR3 are translated into values, which are loaded into KXJCSRF and KXJCSRH, and bit 2 of KXJCSRJ is set.

5. After shared memory is configured and enabled, the KXJ11-CA clears TPR0 and sets TPR1<2:0> = 100. This puts the KXJ11-CA back into the "waiting for command" state.

NOTE

A local reset or a Q-Bus ODT GO command will disable shared memory because it clears KXJCSRJ bit 2. The contents of KXJCSRF and KXJCSRH are unaffected.

To disable shared memory, the following events occur.

1. The arbiter determines that the KXJ11-CA is ready to receive a command. This occurs when the KXJ11-CA is in the "waiting for command" state.

2. The arbiter sets TPR0 bit 8. Only bit 8 should be set. If the arbiter were to set more than one bit at a time in TPR0, an error would result (and would be recorded in TPR1).

3. The KXJ11-CA clears bit 2 of KXJCSRJ, disabling shared memory.

4. The KXJ11-CA clears TPR0 and sets TPR1<2:0> = 100. This puts the KXJ11-CA back into the "waiting for command" state.

3.8.6 Shared Memory Considerations

When designing an application, the user should note the following circumstances under which the use of shared memory could yield unpredictable results.

The KXJ11-CA is designed for use in memory architectures that are non-cached. Arbiters with cache memory (such as the KDJ11) must disable or bypass the cache when accessing shared memory. The KXJ11-CA has no mechanism for updating the arbiter's cache when cached shared memory locations are altered by the on-board J-11 or DMA controller.

CHAPTER 4
DMA TRANSFER CONTROLLER

4.1 OVERVIEW

The DMA transfer controller (DTC) is designed around the AmZ8016 chip. For details on the operation of the AmZ8016 chip, refer to the AmZ8016 DMA Transfer Controller Data Sheet included in this documentation package. The information that follows summarizes and describes the DTC functions implemented on the KXJ11-CA.

The DTC can perform DMA transfers between any of the following addresses.

1. A local address to a local address
2. A local address to a Q-Bus address
3. A Q-Bus address to a local address
4. A Q-Bus address to a Q-Bus address
5. From the channel A transmitter of the multiprotocol SLU through DMA channel 1
6. To the channel A receiver of the multiprotocol SLU through DMA channel 0
7. To/From the PIO chip through DMA channel 1

Word and byte transfers are supported locally. Only word transfers are supported across the Q-Bus. Note that in byte mode, the addressing is the inverse of the PDP-11 addressing scheme. For example, DTC address 1000 corresponds to PDP-11 address 1001, and DTC address 1001 corresponds to PDP-11 address 1000.

The operations of the DTC are controlled by several internal registers. The DTC can load these registers directly from memory, thereby minimizing the amount of processor intervention necessary to perform a DMA transaction. The area of memory where the parameters for the DTC are stored is referred to as the chain table. The local J-11 microprocessor need only load the address of the chain table into the DTC and issue a "start" command to initiate a DMA transfer.

DMA transactions may be initiated locally by either the J-11 or the arbiter CPU. If the transfer is initiated by the arbiter, the command words and transfer parameters are placed in the command registers of the two-port RAM (TPR) file. The local J-11 will then initiate the DMA transaction using the parameters supplied by the arbiter.

The DTC consists of two identical channels. DMA transfers may be interleaved between these two channels or interleaved between the DTC and the J-11. It is also possible to select a "hog mode" that allows the DMA transfer to run to completion without interruption.

The DTC supports three types of operations: Transfer, Search, and Transfer-and-Search. As the name implies, Transfer operations move data from a source to a destination. Search operations read data from a source and compare the data to the pattern register. A mask register allows the user to declare "don't care" bits. The Transfer-and-Search operation combines the features of the Transfer and Search functions. In this type of operation, data is transferred between a source and destination until the data transferred meets the match condition specified in the Channel Mode register.

The DTC is capable of performing multiple DMA transactions without processor intervention. This can be accomplished in two ways: base-to-current reloading or chaining. Base-to-current reloading allows the DTC to reload a portion of its registers before initiating a DMA transfer. The reload operation occurs between internal registers, so there are no memory access related delays. This type of operation is only practical in applications where data is continuously transferred between the same addresses. Chaining allows some or all of the applicable registers of the DTC to be reloaded from a new chain table.

Upon completion of a DMA transfer, the DTC may perform any combination of the following options: interrupt the local processor, perform base-to-current reloading, or perform a chain reload. It may also choose to take no action.

4.2 DATA TRANSFER CONTROLLER (DTC) REGISTERS

NOTE

Refer to Section 4.3 for descriptions of how the DTC registers are used during DMA operations.

The Data Transfer Controller contains two types of registers: global registers and channel registers. Global registers control the overall operation and configuration of the DTC. There are two global registers; the command register and the master mode register (see Table 4-1). Channel registers define the state of a particular channel. There are two identical sets of channel registers, one for each channel. These registers are always accessed as words and are aligned on even (word) address boundaries. Table 4-2 lists the DTC channel registers and their addresses.

Table 4-1 DTC Global Registers

Address	Access	Description
17774454*	W	Command Register
17774470	RW	Master Mode Register

* Location 17774454 can be read or written. When read, it yields status information only. It is written with command information that cannot be read back.

Table 4-2 DTC Channel Registers

Channel 1 Address	Channel 0 Address	Access	Description
17774400	17774402	RWC	Current B Address Offset
17774404	17774406	RWC	Base B Address Offset
17774410	17774412	RWC	Current A Address Offset
17774414	17774416	RWC	Base A Address Offset
17774420	17774422	RWC	Current B Address Segment/Tag
17774424	17774426	RWC	Base B Address Segment/Tag
17774430	17774432	RWC	Current A Address Segment/Tag
17774434	17774436	RWC	Base A Address Segment/Tag
17774440	17774442	RWC	Chain Address Offset
17774444	17774446	RWC	Chain Address Segment/Tag
17774450	17774452	R	Interrupt Save Register
17774454*	17774456	R	Status Register
17774460	17774462	RWC	Current Operation Count
17774464	17774466	RWC	Base Operation Count
	17774472	X	Reserved
17774474	17774476	X	Reserved
17774500	17774502	X	Reserved
17774504	17774506	X	Reserved
17774510	17774512	RWC	Pattern Register
17774514	17774516	RWC	Mask Register
17774520	17774522	RWC	Channel Mode Low
17774524	17774526	RWC	Channel Mode High
17774530	17774532	RWC	Interrupt Vector
17774534	17774536	X	Reserved

* Location 17774454 can be read or written. When read, it yields status information only. It is written with command information that cannot be read back.

The KXJ11-CA DTC is based on the AmZ8016 chip (as described in the AmZ8016 Technical Manual). Several registers of the AmZ8016 chip are not implemented in the KXJ11-CA. These are shown as "reserved" in the tables.

The tables specify the access code for each register. The key to the abbreviations used is as follows.

R = The register can be read by the on-board J-11 processor.

W = The register can be written by the on-board J-11 processor.

C = The register can be loaded by the DTC as part of a chaining operation.

X = The register is not implemented or is reserved for future use by Digital Equipment Corp.

4.2.1 DTC Global Registers

4.2.1.1 Command Register -- The Command Register is the write-only register that the on-board J-11 uses to issue commands to the DTC. These commands include Reset, Start Chain, and others (see Figure 4-1).

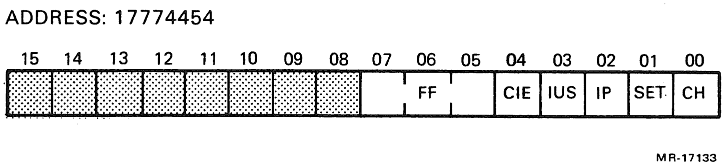


Figure 4-1 DTC Command Register

Bits	Name	Description
15:8		Not used
7:5	FF	Function field - specifies the type of function to be performed by the DTC.

FF	Function
000	Reset
001	Interrupt
010	Software Request
011	Flip Bit
100	Hardware Mask
101	Start Chain
110	Reserved
111	Reserved

Bits	Name	Description
4	CIE	Channel interrupt enable - When set, indicates that interrupt requests are enabled.

Bits	Name	Description
3	IUS	Interrupt under service - When set, indicates that an interrupt is being serviced.
2	IP	Interrupt pending - When set, indicates that an interrupt request is currently pending.
1	SET	Set/Clear - When set, specifies a set or 1 condition. When cleared, specifies a clear or 0 condition.
0	CH	Channel 0/Channel 1 - When set, specifies channel 1. When cleared, specifies channel 0.

Table 4-3 summarizes the functions that can be performed by writing the various bits of the DTC Command Register.

Table 4-3 DTC Command Summary

	DTC Command Register Bits		
Command	76	543	210
Reset	00	0XX	XXX
Start Chain Channel 0	10	1XX	XX0
Start Chain Channel 1	10	1XX	XX1
Set Software Request Channel 0	01	0XX	X10
Set Software Request Channel 1	01	0XX	X11
Clear Software Request Channel 0	01	0XX	X00
Clear Software Request Channel 1	01	0XX	X01
Set Hardware Mask Channel 0	10	0XX	X10
Set Hardware Mask Channel 1	10	0XX	X11
Clear Hardware Mask Channel 0	10	0XX	X00
Clear Hardware Mask Channel 1	10	0XX	X01
Set CIE, IUS, IP Channel 0	00	1ES	P10
Set CIE, IUS, ID Channel 1	00	1ES	P11
Clear CIE, IUS, IP Channel 0	00	1ES	P00
Clear CIE, IUS, IP Channel 1	00	1ES	P01
Set Flip Bit Channel 0	01	1XX	X10
Set Flip Bit Channel 1	01	1XX	X11
Clear Flip Bit Channel 0	01	1XX	X00
Clear Flip Bit Channel 1	01	1XX	X01

Notes:

E = Set to perform set/clear on CIE, clear for no effect on CIE.
AS = Set to perform set/clear on IUS, clear for no effect on IUS.
P = Set to perform set/clear on IP, clear for no effect on IP.
X = "Don't care" bit. This bit is not decoded and may be 0 or 1.

4.2.1.2 **Master Mode Register** -- The Master Mode Register controls various aspects of overall DTC operation (see Figure 4-2).

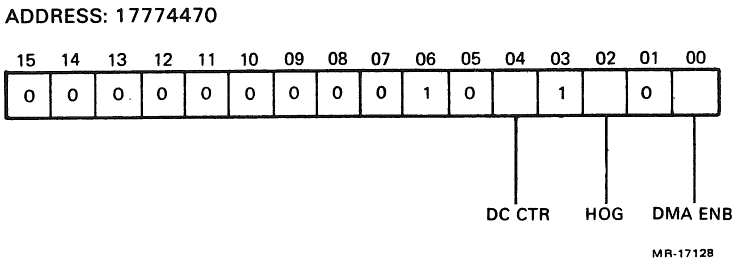


Figure 4-2 Master Mode Register

Bits	Name	Description
15:8		Not used (read as ones)
7		Must be zero
6		Must be one
5		Must be zero
4	DC CTR	Daisy chain control - When set, inhibits interrupt requests from the on-board PIO counter/timer. The PIO counter/timer is part of an interrupt daisy chain at a higher level than the DTC.
3		Must be one.
2	HOG	Hog mode - When set, the DTC interleaves control of the local I/O bus with the on-board J-11. When cleared, the DTC retains control of the bus until a terminal condition exists (as indicated by the contents of the Current Operation Count Register described in Section 4.2.2.6). This is also called "hog mode".
1		Must be zero
0	DMA ENB	DMA enable - When set, allows the DTC to request control of the local I/O bus. When cleared, prevents chaining or DMA operations.

4.2.2 DTC Channel Registers

4.2.2.1 Current Address Registers A and B -- Each channel has two Current Address Registers; one that specifies the current source address of a DTC transfer and one that specifies the current destination address. The "flip bit" in the Channel Mode Register (see Section 4.2.2.8) specifies which registers (A or B) are the source and which registers are the destination. A complete Current Address Register consists of two words, a segment/tag and an offset. The segment/tag specifies.

- If the source (or destination) address resides on the Q-Bus
- If the source (or destination) address resides in the I/O page
- Address bits <21:16> of the source (or destination) address
- If the source (or destination) address should be incremented, decremented, or held constant as the transfer proceeds

The segment/tag has the following format (Figure 4-3).

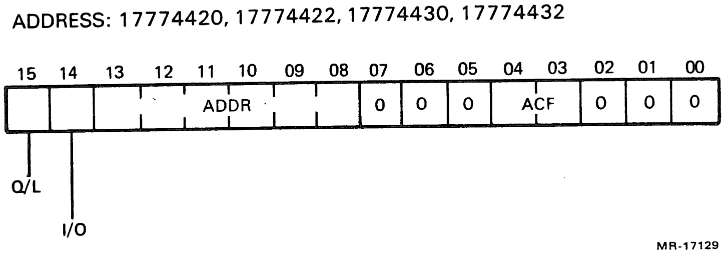


Figure 4-3 Current A or B Address Segment/Tag

Bits	Name	Description
15	Q/L	Bus Choice - When set, causes the current source (or destination) address to reside on the Q-Bus. When the bit is cleared, the current address is a local address (KXJ11-CA).
14	I/O	I/O bit - When set, causes the Q-Bus signal BBS7 to be asserted, which forces a reference to the I/O page. The referenced I/O page can reside locally (if bit 15 is cleared) or on the Q-Bus (if bit 15 is set).
13:8	ADDR	Bits <21:16> of the current address.
7:5		Must be zero

Bits	Name	Description
4:3	ACF	Count method - determines how addresses will be affected as the DTC transfer proceeds. **ACF** **Function** 00 Increment address 01 Decrement address 10 Hold address 11 Hold address
2:0		Must be zero

The offset consists of bits <15:00> of the source (or destination) address (Figure 4-4).

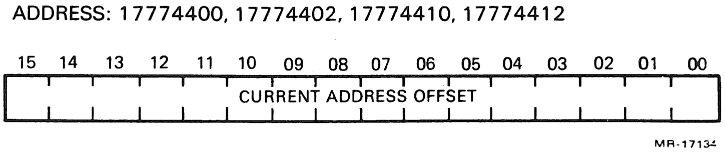


Figure 4-4 Current A or B Address Offset

4.2.2.2 Base Address Registers A and B -- The formats of Base Address Registers A and B are identical to those of Current Address Registers A and B. At the beginning of a transfer, the Base Address Registers and Current Address Registers are loaded with the same information. A transfer can be restarted by reloading the Current Address Registers with the contents of the Base Address Registers. Refer to Figures 4-3 and 4-4 for the register bit descriptions.

4.2.2.3 Chain Address Register -- The Chain Address Register is used to point to a "reload word", the first word in a chain table (see Section 4.3.1). The reload word specifies which registers are to be loaded, in order to set up a chaining operation. The other chain table entries contain the data with which the registers are loaded.

The Chain Address Register consists of two words, a segment/tag and an offset. The segment/tag specifies:

- If the reload word address resides on the Q-Bus
- If the reload word address resides in the I/O page
- Address bits <21:16> of the reload word address

The segment/tag has the following format (Figure 4-5).

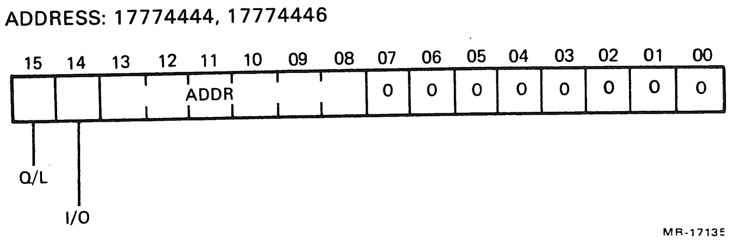


Figure 4-5 Chain Address Segment/Tag

Bits	Name	Description
15	Q/L	Bus Choice - When set, indicates that the reload word address resides on the Q-Bus. When cleared, indicates that the address is a local (KXJ11-CA) one.
14	I/O	I/O bit - When set, causes the Q-Bus signal BBS7 to be asserted, which forces a reference to the I/O page. The referenced I/O page can reside locally (if bit 15 is cleared) or on the Q-Bus (if bit 15 is set).
13:8	ADDR	Bits <21:16> of the reload word address.
7:0		Must be zeros

The offset consists of bits <15:0> of the reload word address (Figure 4-6).

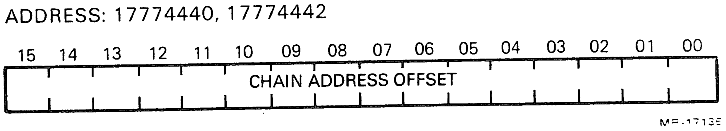


Figure 4-6 Chain Address Offset

4.2.2.4 Interrupt Vector and Interrupt Save Register -- Each channel has an Interrupt Vector Register and an Interrupt Save Register. The Interrupt Vector Register contains the vector that is output during an interrupt acknowledge cycle. When an interrupt occurs, the contents of the Interrupt Vector Register and part of the Status Register are loaded automatically into the Interrupt Save Register. This allows a new vector to be loaded during chaining and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. The Interrupt Save Register can be read but can not be directly written by the user.

The Interrupt Vector Register has the following format (Figure 4-7).

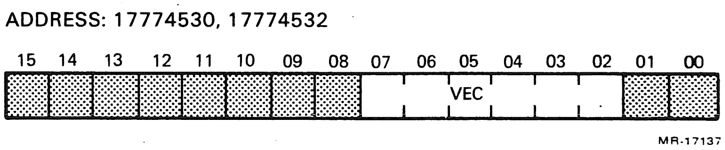


Figure 4-7 Interrupt Vector Register

Bits	Name	Description
15:8		Not used (read/write)
7:2	VEC	Interrupt vector
1:0		Not used (read/write)

The Interrupt Save Register has the following format (Figure 4-8).

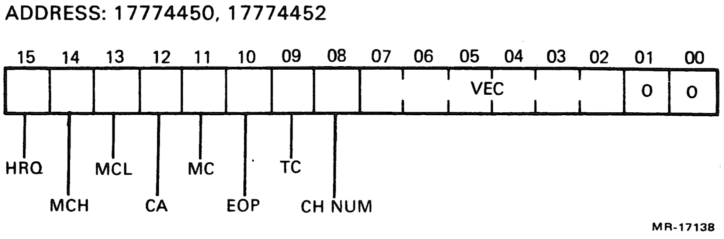


Figure 4-8 Interrupt Save Register

Bits	Name	Description
15	HRQ	Hardware request - A copy of Status Register bit 5.
14	MCH	Match count high - A copy of Status Register bit 4.

Bits	Name	Description
13	MCL	Match count low - A copy of Status Register bit 3.
12	CA	Chain abort - A copy of Status Register bit 12.
11	MC	Match count - A copy of Status Register bit 2.
10	EOP	End of process - A copy of Status Register bit 1.
9	TC	Terminal count - A copy of Status Register bit 0.
8	CH NUM	Channel number - When set, refers to channel 1. When cleared, refers to channel 0.
7:2	VEC	Interrupt vector - A copy of Interrupt Vector Register bits <7:2>.
1:0		Not used

4.2.2.5 Status Register -- Each channel has a read-only Status Register. Each status register contains an interrupt status field (bits <15:13>), a DTC status field (bits <12:9>), a hardware interface field (bits <6:5>), and a completion status field (bits <4:0>). The bits that comprise these fields are described below. Parts of the status register are copied to the Interrupt Save Register when an interrupt occurs. Note that bits <12:9> = 0000 indicate that the channel is initialized and waiting for a request.

The Status Register has the following format (Figure 4-9).

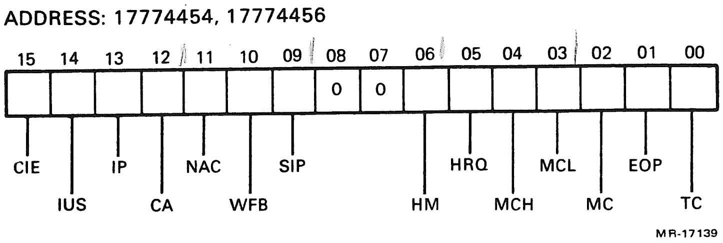


Figure 4-9 Status Register

Bits	Name	Description
15	CIE	Channel interrupt enable - When set, indicates that interrupt requests are enabled. Set the same as bit 4 of the Command Register.
14	IUS	Interrupt under service - When set, indicates that an interrupt is being serviced. Set the same as bit 3 of the Command Register.
13	IP	Interrupt pending - When set, indicates that an interrupt request is currently pending. Set the same as bit 2 of the Command Register.
12	CA	Chain abort - When set, indicates that a chaining operation has been terminated. This bit is also set when the DTC is initialized. This bit is cleared when a new chain address segment/tag or offset word is loaded.
11	NAC	No auto reload on chaining - When set, indicates that the channel has completed a DMA transfer and that neither base-to-current reloading nor auto-chaining were enabled. This bit is also set when the DTC is initialized. This bit is cleared when a Start Chain Command is issued.
10	WFB	Waiting for bus - When set, indicates that the channel wants control of a bus to perform a DMA transfer.
9	SIP	Second interrupt pending - When set, indicates that a second interrupt is pending on the channel, and that channel activity should be suspended until an interrupt acknowledge occurs.
8:7		Must be zero
6	HM	Hardware mask - When set, indicates that this channel is inhibited from responding to the assertion of the channel's hardware request line.
5	HRQ	Hardware request - When set, indicates that the channel's hardware request line is asserted.

Bits	Name	Description
4	MCH	Match count high - When set, indicates a match between the upper byte of data being transferred-and-searched or searched, and the pattern determined by the Pattern and Mask Registers.
3	MCL	Match count low - When set, indicates a match between the lower byte of data being transferred-and-searched or searched, and the pattern determined by the Pattern and Mask Registers.
2	MC	Match count - When set, indicates that a DMA operation was terminated due to a match between data being transferred-and-searched or searched, and the condition specified by bits <1:0> of the Channel Mode High Register.
1	EOP	End of process - When set, indicates that a DMA operation was terminated due to the assertion of the DTC's end of process (EOP) line.
0	TC	Terminal count - When set, indicates that a DMA operation was terminated because the operation count reached zero.

4.2.2.6 Current and Base Operation Count Registers -- Each channel has a Current Operation Count Register, which specifies the number of words (or bytes) remaining to be transferred for a DMA operation. The contents of the register are decremented by one each time a datum is transferred. A DMA operation can be resumed where it left off by using the count contained in this register. When a DMA transfer is completed, the register contains zero. The maximum count (64 K) is specified by loading this register with zero.

Each channel also has a Base Operation Count Register. The contents of the Base Operation Count Register are initially identical to those of the Current Operation Count Register. As the transfer progresses, however, the contents of the Base Operation Count Register are not decremented. If a DMA transfer needs to be restarted from scratch, the original byte (or word) count can be restored by loading the contents of the Base Operation Count Register into the Current Operation Count Register.

Refer to Figures 4-10 and 4-11 for the register formats.

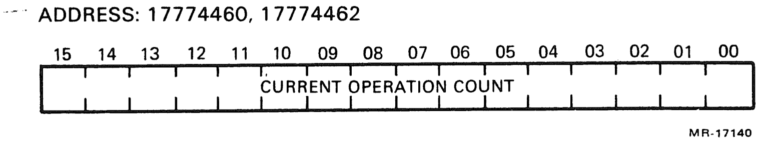


Figure 4-10 Current Operation Count Register

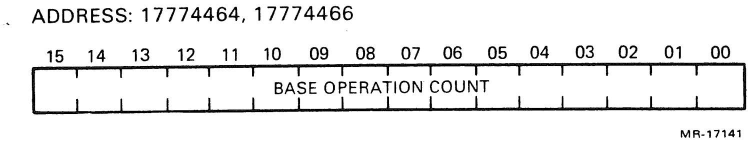


Figure 4-11 Base Operation Count Register

4.2.2.7 Pattern and Mask Registers -- Each channel has a Pattern Register and a Mask Register, which are used in search and transfer-and-search operations. The Pattern Register contains a pattern that read data is compared with to determine whether or not a "match" condition exists. The user can program the DTC to stop a search when there is a match or when there is no match. The Mask Register is used to exclude selected bits from the comparison. Setting a Mask Register bit to "1" excludes that bit from the comparison. The formats of the Pattern and Mask Registers are shown in Figures 4-12 and 4-13.

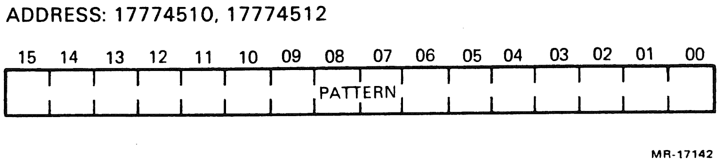


Figure 4-12 Pattern Register

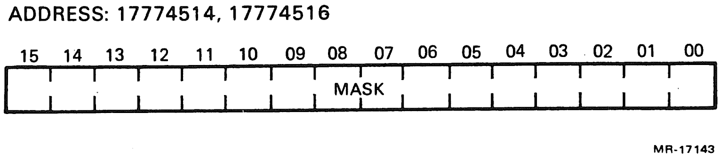


Figure 4-13 Mask Register

4.2.2.8 Channel Mode Register -- Each channel has a Mode Register. The Mode Register specifies what type of DMA operation a channel will perform, how the operation will be executed, and what action, if any, will be taken when the operation is completed. The mode register consists of two words: a Channel Mode High and a Channel Mode Low. Channel Mode High is used to

- Initiate a DMA operation
- Specify what is to occur if a match condition exists
- Determines how software and hardware requests are handled.

Channel Mode High has the following format (Figure 4-14).

ADDRESS: 17774524, 17774526

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	0	0	0	0	SR		0	MC	

MR-17144

Figure 4-14 Channel Mode High

Bits	Name	Description
15:5		Not used (read as zeros)
4	SR	Software request - When set, initiates a DMA operation. The channel requests the bus and performs transfers as specified by XFER in Channel Mode Low.
3	HM	Hardware mask - When set, prevents the channel from responding to the assertion of the channel's hardware request line.
2		Not used (read as zero)
1:0	MC	Match condition - Specifies what will occur for match conditions.

MC	Action
00	Stop on no match
01	Stop on no match
10	Stop on word match
11	Stop on byte match

Channel Mode Low specifies

- The type of operation and transfer performed
- Which of the Current Address Registers (A or B) is the source and which is the destination
- What will occur when a DMA operation is completed

Channel Mode Low has the following format (Figure 4-15).

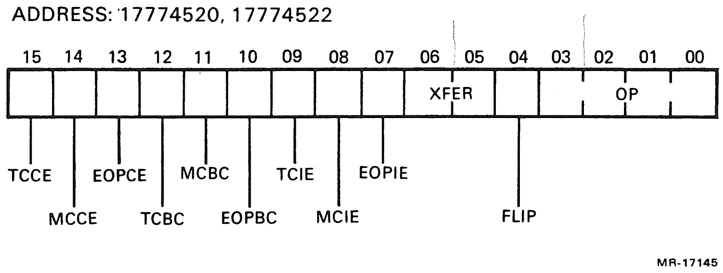


Figure 4-15 Channel Mode Low

Bits	Name	Description
15	TCCE	Terminal count chain enable - When set, causes a chain reload for the next DMA operation if the Current Operation Count Register is decremented to zero.
14	MCCE	Match count chain enable - When set, causes a chain reload for the next DMA operation if a match condition exists.
13	EOPCE	End of process chain enable - When set, causes a chain reload for the next DMA operation if an end-of-process (EOP) termination occurs.
12	TCBC	Terminal count base-current - When set, causes a base-to-current reload if the Current Operation Count Register is decremented to zero.
11	MCBC	Match count base-current - When set, causes a base-to-current reload if a match condition exists.
10	EOPBC	End of process base-current - When set, causes a base-to-current reload if an end-of-process (EOP) termination occurs.
9	TCIE	Terminal count interrupt enable - When set, the channel issues an interrupt if the Current Operation Count Register is decremented to zero.

Bits	Name	Description
8	MCIE	Match count interrupt enable - When set, the channel issues an interrupt if a match condition exists.
7	EOPIE	End of process interrupt enable - When set, the channel issues an interrupt if an end-of-process (EOP) termination occurs.
6:5	XFER	Transfer type - Specifies the type of transfer the channel will perform. Refer to the DTC Technical Manual for descriptions of these transfers.

XFER	Transfer Type
00	Single transfer
01	Demand dedicated with bus hold
10	Demand dedicated with bus release
11	Channel to channel demand interleave

Bits	Name	Description
4	FLIP	Flip bit - When set, Current Address Register B contains the source, and Current Address Register A contains the destination of a transfer. When cleared, Current Register A contains the source and Current Address Register B contains the destination.
3:0	OP	Operation type - Specifies the type of operation the channel will perform. See the DTC Technical Manual for descriptions of these operations.

		Operand Size		Transaction
OP	Operation	A	B	Type
0000	Transfer	Word	Word	Flowthrough
0001	Transfer	Byte	Byte	Flowthrough
0010	Reserved			
0011	Reserved			
0100	Trnsf-Search	Word	Word	Flowthrough
0101	Trnsf-Search	Byte	Byte	Flowthrough
0110	Reserved			
0111	Reserved			
1000	Transfer	Byte	Word	Flowthrough
1001	Reserved			
1010	Reserved			
1011	Reserved			
1100	Trnsf-Search	Byte	Word	Flowthrough
1101	Reserved			
1110	Search	Word	Word	Read
1111	Search	Byte	Byte	Read

NOTE

"Flyby" operations are not supported on the KXJ11-CA.

4.3 PROGRAMMING THE DTC

Programming the DTC consists of three phases: Chip Initialization, DMA Operation, and Termination. This section will provide a general description of these phases.

4.3.1 Chip Initialization

The RESET instruction is used to place the DTC in a known state. A reset will clear the CIE, IP, SIP and WFB bits, and set the CA and NAC bits in the Channel Status registers. The Master Mode register will also be cleared. Before a DMA operation is initiated, the local CPU loads the Master Mode register and the Chain Address register of the appropriate channel of the DTC. The DTC fetches any other parameters that are necessary from a table located in system memory known as the chain table. This minimizes the amount of CPU intervention necessary to perform a DMA operation. The relationship of the Chain Address Register to the chain table is shown in Figure 4-16.

The first word in the chain table is the reload word. The reload word is used to specify which registers should be loaded for the pending DMA operation. Bits <9:0> of the reload word correspond to the registers of the DTC as shown in Figure 4-17. Bits <15:10> are not used.

Therefore, if a bit in the reload word is set, then the corresponding registers are to be reloaded from the chain table. Since all of the registers are not applicable to each DMA operation, the chain table may be of variable length (that is, the pattern and mask registers would not be used in DMA operations that do not search the data). It is incorrect to select a register in the reload word and subsequently load that register with a dummy argument such as zero. Figures 4-18 and 4-19 show examples of the relationship between the reload word and the chain table.

The DTC has been properly initialized once the chain tables have been created, and the Master Mode register and Chain Address Register for the selected channel have been loaded.

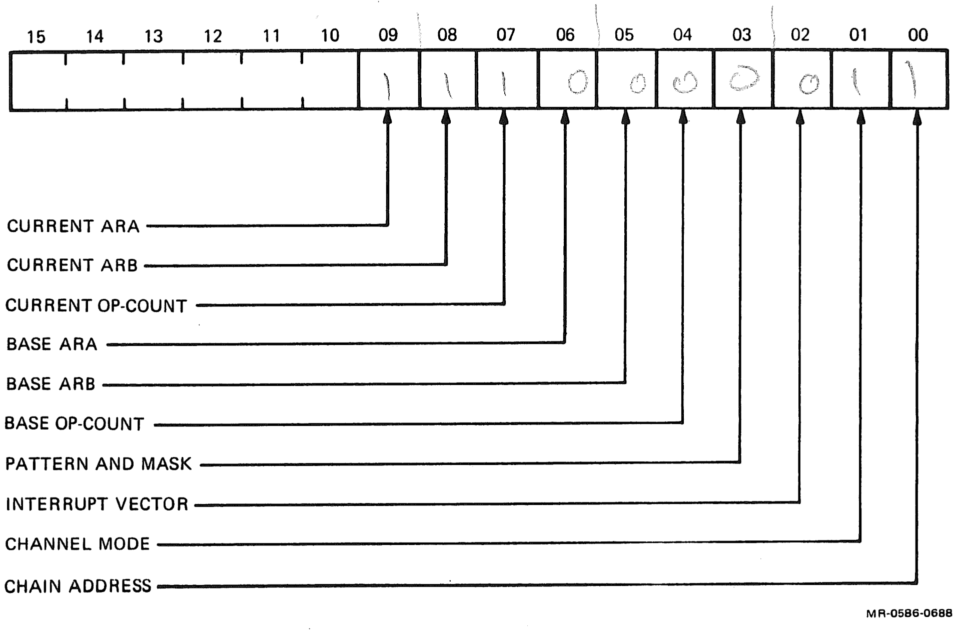


Figure 4-16   Chain Address Register

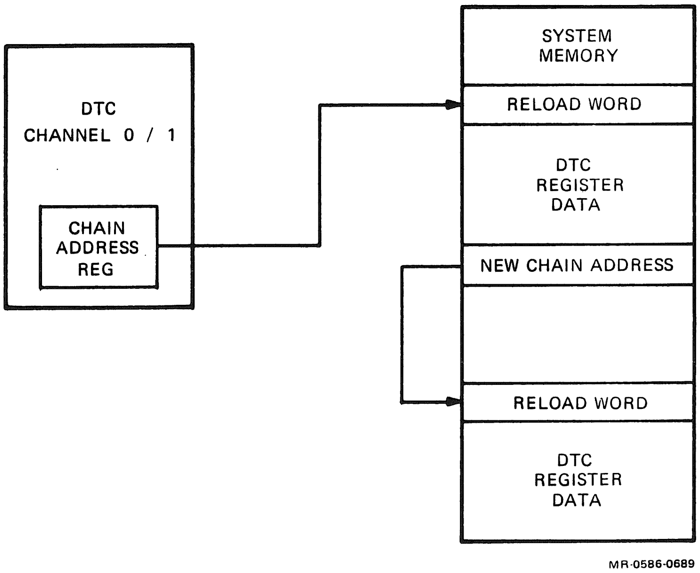


Figure 4-17   Reload Word

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
X	X	X	X	X	X	1	1	1	1	0	0	0	0	1	0

CURRENT ARA SEGMENT/TAG
CURRENT ARA OFFSET
CURRENT ARB SEGMENT/TAG
CURRENT ARB OFFSET
CURRENT OP—COUNT
CHANNEL MODE HIGH
CHANNEL MODE LOW

MR-0186-0098

Figure 4-18 Example 1

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
X	X	X	X	X	X	1	0	1	0	0	0	1	0	1	1

CURRENT ARA SEGMENT/TAG
CURRENT ARA OFFSET
CURRENT OP—COUNT
PATTERN REGISTER
MASK REGISTER
CHANNEL MODE HIGH
CHANNEL MODE LOW
CHAIN ADDRESS SEGMENT/TAG
CHAIN ADDRESS OFFSET

MR-0186-0099

Figure 4-19 Example 2

4.3.2 DMA Operation

The DTC may perform a DMA operation once it has been properly initialized. A DMA operation may be initiated in one of four ways: by software request, by hardware request, by loading a set software request bit in the Channel Mode register during chaining, or as the result of a command from the arbiter.

Software Request: The local CPU may initiate a DMA operation by writing a Start Chain command to the Command Register. If the Software Request bit is not set as part of the start chain command, then the Software Request command can be issued to initiate the DMA operation. The Software Request command sets the software request bit in the channel's mode register. If either the Second Interrupt Pending (SIP) bit or the No Auto-Reload or Chain (NAC) bit is set in the channel's status register, the DMA operation will not begin. The SIP bit will be cleared when the channel receives an interrupt acknowledge. The NAC bit will be cleared when the channel receives a Start Chain command. The Start Chain command initiates the DMA operation after the registers of the selected channel are loaded from the chain table. The Start Chain command is ignored if the SIP bit or the Chain Abort (CA) bit are set in the channel's status register. The SIP bit was described above. The CA bit is cleared when the channel's chain address register is reloaded.

Hardware Request: DMA operations may be started by asserting a channel's DREQ input from SLU2 or the PIO. The mask bit in the Channel Mode Register controls whether or not this request is detected.

Starting After Chaining: If the software request bit of the channel's Master Mode register is set during chaining, the channel will perform the DMA operation at the end of chaining.

Arbiter Request: The arbiter may interrupt the local CPU to request a DMA operation. This is accomplished by passing parameters to load the chain address register of channel 0 through the two-port RAM. The arbiter loads register 2 of the TPR with the offset of the chain address register, and loads register 3 of the TPR with the segment/tag of the chain address register. The DMA operation is then initiated by setting the DMA Load bit (bit 1) in the TPR command register (register 0). Error conditions will be returned in TPR register 1.

Information in the channel's mode register determines what type of DMA operation will be performed. The Channel Mode register consists of two words: Channel Mode High and Channel Mode Low.

Bits <3:0> of the Channel Mode Low register select the type of DMA operation. These bits determine whether the data should be transferred, searched, or transferred-and-searched. Bit 4 is the flip bit. It is used to determine which set of current address registers (CARA, CARB) points to the source.

Bits <6:5> determine the transfer type. The types of DTC transfers are single transfer, demand dedicated with bus hold, demand dedicated with bus release, and channel-to-channel demand interleave. Single transfer is used with devices that transfer data at irregular intervals. A single DMA transaction will occur each time a Software Request command is issued or the DREQ input is asserted. Demand dedicated with bus hold is a software hog mode. This mode allows the DMA transaction to run to completion for local addresses, as long as there is a valid op count and the DREQ input is asserted. If the DREQ input is not asserted, no DMA operations will occur but the channel will retain bus control. In Q-Bus hog mode, the KXJ11-CA releases the bus and requests the bus again after each word transfer. Demand dedicated with bus release is similar to demand dedicated with bus hold, because a DMA transaction is allowed to run to completion if DREQ is asserted. If DREQ is not asserted, the DTC must release the bus, thus allowing other devices to obtain the bus. The operation performed by a channel-to-channel demand interleave request depends on the state of bit 2 in the Master Mode register. If MM bit 2 is clear, then control may be passed between each channel of the DTC without the need to release the bus. If MM bit 2 is set, then the DTC must share the bus with the local processor. The DTC will release the bus and then request it again after every DMA iteration.

Bits <1:0> of the Channel Mode High register are used to determine the type of match control in Search and Transfer-and-Search operations. The DTC is capable of generating a termination condition based on 'No Match', 'Word Match', and 'Byte Match'.

Bit <4> of the Channel Mode High register causes the channel to request the bus and perform transfers when it is set by a Software Request command or a chain reload.

4.3.3 Termination Options

Bits <15:7> of the Channel Mode Low register control the termination options. A DTC operation may be terminated in a number of ways. If the Current Operation Count Register goes to zero, then a Terminal Count (TC) termination is generated. External logic may assert the End Of Process (EOP) input of the DTC to generate an EOP termination at any time. In addition, during a Search or Transfer-and-Search operation, a match condition may occur that generates a MC termination. Bits <15:7> allow the DTC to perform a chain reload, a base-to-current reload, or to interrupt the local processor if a TC, EOP, or MC termination condition is encountered. If bits <15:7> are cleared, then no special action is initiated when a TC, EOP, or MC condition is encountered.

4.3.3.1 EOP Condition Handling

EOP conditions occur either as a result of a reference to nonexistent memory or due to a Q-Bus timeout condition. Q-Bus timeouts occur when the KXJ11-CA is not granted access to the Q-Bus by the arbiter within a fixed period of time (approximately 140 usecs). The condition only occurs during periods of heavy contention for the Q-Bus by other peripherals performing DMA operations. It is possible to recover the DMA transfer operation from the point of termination, depending upon the application of the DMA channel.

To differentiate between an EOP condition resulting from a Q-Bus timeout and a reference to nonexistent memory, bit 5 of KXJCSRJ (SACK TOUT) may be tested. This status bit is used to notify the application that a Q-Bus timeout condition has occurred. It is a global status bit because it is not specific to a particular DMA channel. This bit may be set by a timeout condition that occurred on one or both DMA channels. We suggest that the exception condition handler always have access to the needed context of both DMA channels.

Once it has been determined that a Q-Bus timeout condition has occurred, it is possible to restart the DMA operation by using the Current Operation Count register (Section 4.2.2.6). This register contains the number of words (or bytes, depending on mode of operation) remaining to be transferred.

4.3.4 Examples

The following example programs were developed on a PDP-11/23+ system with 256KB of memory, using the RT-11 (version 5.1) operating system with the KXJ11-CA Peripheral Processor Software Toolkit. The programmer should be familiar with MACRO-11 and the KXJ11-CA Peripheral Processor Toolkit.

```
.TITLE EXAM1.MAC

; This program transfers data from local KXJ11-CA addresses to
; local KXJ11-CA addresses. This program should be compiled and
; linked on the development system, and then downloaded into the
; KXJ11-CA using the KXJ11-CA Software Toolkit. Once the program
; has been compiled and linked, use the following KUI commands to
; execute it and verify its success.
;
; .KUI
; KUI>SET n                 ! Where n is the appropriate KXJ11-CA
; KUI>LOAD EXAM1
; KUI>ODT                   ! Use KUI ODT to verify that the
;       .                     destination addresses are cleared
;       .
; ODT>CTRL/C
; KUI>EXECUTE               ! Execute EXAM1
; KUI>ODT                   ! Use KUI ODT to verify that the transfer
;       .                     was successful
;       .
; ODT>CTRL/C
; KUI>EXIT
;

; SET UP REGISTER ASSIGNMENTS

MMREG   =       174470  ; MASTER MODE REGISTER

CMDREG  =       174454  ; COMMAND REGISTER
CASTF0  =       174446  ; CHANNEL 0 CHAIN ADDRESS SEGMENT/TAG
                        ; FIELD
CAOF0   =       174442  ; CHANNEL 0 CHAIN ADDRESS OFFSET FIELD

START: MOVB #130,MMREG    ; LOAD MASTER MODE REG TO DISABLE DTC

       CLRB CMDREG        ; RESET THE DTC

       MOV  #0,CASTF0     ; LOAD THE CHAIN ADDRESS REGISTER
                          ; SEG/TAG
       MOV  #RELOAD,CAOF0 ; LOAD THE CHAIN ADDRESS REGISTER OFFSET

       MOVB #131,MMREG    ; LOAD MASTER MODE REG TO ENABLE DTC

       MOVB #102,CMDREG   ; SET SOFTWARE REQUEST CHANNEL 0

       MOVB #240,CMDREG   ; START CHAIN CHANNEL 0

       BR      .          ; STAY HERE WHILE THE USER VERIFIES
                          ; THAT THE PROGRAM WAS SUCCESSFUL

; CHAIN LOAD REGION

RELOAD: .WORD 001602      ; RELOAD WORD <Select CARA,CARB,COPC,CM>

.WORD   000000            ; CURRENT ADDRESS REGISTER A SEG/TAG
.WORD   SOURCE            ; CURRENT ADDRESS REGISTER A OFFSET
                          ; <This local address is the source>

.WORD   000000            ; CURRENT ADDRESS REGISTER B SEG/TAG
.WORD   DESTNT            ; CURRENT ADDRESS REGISTER B OFFSET
                          ; <This local address is the
                          ; destination>

.WORD   000013.           ; CURRENT OPERATION COUNT <Transfer 13
                          ; words>

.WORD   000000            ; CHANNEL MODE REGISTER HIGH
.WORD   000040            ; CHANNEL MODE REGISTER LOW
                          ; <No match conditions, do nothing
                          ; upon completion, transfer
                          ; type = Demand Dedicated w/Bus Hold,
                          ; CARA = source, word transfers>
```

```
SOURCE: .WORD   1,2,3,4,5,6,7,6,5,4,3,2,1

DESTNT: .BLKW   13.

        .END START

.TITLE EXAM2.MAC

; This program transfers data from local KXJ11-CA addresses to
; global Q-Bus addresses. This program should be compiled and
; linked on the development system, and then downloaded into the
; KXJ11-CA using the KXJ11-CA Software Toolkit. Once the program
; has been compiled and linked, use the following commands to
; execute it and verify its success.
;
; <HALT the development machine so that locations may be examined
;  with Q-Bus ODT>
; @600000/xxxxxx   ! Examine the destination locations and clear
;     .              them if necessary
;         .
; @600030/xxxxxx
; @P               ! Use the 'P' command to return to the system
                     prompt
;
; .KUI
; KUI>SET n        ! Where n is the appropriate KXJ11-CA
; KUI>LOAD EXAM2
; KUI>EXECUTE
; KUI>EXIT
;
; <HALT the development machine so that locations may be examined
;  with Q-Bus ODT>
; @600000/xxxxxx   ! Examine the destination locations to verify
;         .          the success of the transfer
;         .
; @600030/xxxxxx
;
; SET UP REGISTER ASSIGNMENTS

MMREG   =   174470     ; MASTER MODE REGISTER
CMDREG  =   174454     ; COMMAND REGISTER
CASTF0  =   174446     ; CHANNEL 0 CHAIN ADDRESS SEGMENT/TAG FIELD
CAOF0   =   174442     ; CHANNEL 0 CHAIN ADDRESS OFFSET FIELD

START:  MOVB  #130,MMREG       ; LOAD MASTER MODE REG TO DISABLE DTC

        CLRB   CMDREG          ; RESET THE DTC

        MOV    #0,CASTF0       ; LOAD THE CHAIN ADDRESS REGISTER
                               ; SEG/TAG
        MOV    #RELOAD,CAOF0   ; LOAD THE CHAIN ADDRESS REGISTER
                               ; OFFSET
        MOVB   #131,MMREG    ; LOAD MASTER MODE REG TO ENABLE DTC

        MOVB   #102,CMDREG   ; SET SOFTWARE REQUEST CHANNEL 0

        MOVB   #240,CMDREG   ; START CHAIN CHANNEL 0

        BR    .              ; STAY HERE WHILE THE USER VERIFIES
                             ; THAT THE PROGRAM WAS SUCCESSFUL
; CHAIN LOAD REGION

RELOAD: .WORD 001602    ; RELOAD WORD <Select CARA,CARB,COPC,CM>

        .WORD  000000   ; CURRENT ADDRESS REGISTER A SEG/TAG
        .WORD  SOURCE   ; CURRENT ADDRESS REGISTER A OFFSET
                        ; <This local address is the source>

        .WORD  101400   ; CURRENT ADDRESS REGISTER B SEG/TAG
        .WORD 00000     ; CURRENT ADDRESS REGISTER B OFFSET
                        ; <This global Q-Bus address is the
                        ; destination>
                        ; <This corresponds to address 600000 on
                        ; the Q-Bus>
                        ; <The DTC uses physical addresses only>

        .WORD  000013.  ; CURRENT OPERATION COUNT <Transfer 13
                        ; words>

        .WORD  000000   ; CHANNEL MODE REGISTER HIGH
        .WORD  000040   ; CHANNEL MODE REGISTER LOW
                        ; <No match conditions, do nothing upon
                        ; completion, transfer type = Demand
                        ; Dedicated w/Bus Hold,
                        ; CARA = source, word transfers>

SOURCE: .WORD  1,2,3,4,5,6,7,6,5,4,3,2,1

        .END START
```

```
.TITLE EXAM3.MAC

; This program transfers data from global Q-Bus addresses to local
; KXJ11-CA addresses.  This program should be compiled and linked
; on the development system, and then downloaded into the KXJ11-CA
; using the KXJ11-CA Software Toolkit.  Once the program has been
; compiled and linked, use the following commands to execute it
; and verify its success.
;
; <Use Q-Bus ODT to deposit values in locations
; 600000(8)-->600030(8).
; These values will be the source for this operation>
;
; @600000/000001   ! Deposit source values
;           .
;           .
; @600030/000001
; @P               ! Use the 'P' command to return to the system
                     prompt
;
; .KUI
; KUI>SET n        ! Where n is the appropriate KXJ11-CA
; KUI>LOAD EXAM3
; KUI>EXECUTE
; KUI>ODT          ! Use KUI ODT to examine the destination
;                    locations to verify that the transfer was
;                    successful
; ODT> .
;      .
;      .
; ODT>CTRL/C
; KUI>EXIT
;
; SET UP REGISTER ASSIGNMENTS

MMREG   =   174470    ; MASTER MODE REGISTER
CMDREG  =   174454    ; COMMAND REGISTER
CASTF0  =   174446    ; CHANNEL 0 CHAIN ADDRESS SEGMENT/TAG FIELD
CAOF0   =   174442    ; CHANNEL 0 CHAIN ADDRESS OFFSET FIELD

START:  MOVB  #130,MMREG     ; LOAD MASTER MODE REG TO DISABLE DTC

        CLRB  CMDREG         ; RESET THE DTC

        MOV   #0,CASTF0      ; LOAD THE CHAIN ADDRESS REGISTER
                             ; SEG/TAG
        MOV   #RELOAD,CAOF0  ; LOAD THE CHAIN ADDRESS REGISTER
                             ; OFFSET

        MOVB  #131,MMREG     ; LOAD MASTER MODE REG TO ENABLE DTC
        MOVB  #102,CMDREG    ; SET SOFTWARE REQUEST CHANNEL 0

        MOVB  #240,CMDREG    ; START CHAIN CHANNEL 0

        BR                   ; STAY HERE WHILE THE USER VERIFIES
                             ; THAT THE PROGRAM WAS SUCCESSFUL
; CHAIN LOAD REGION

RELOAD:.WORD 001602   ; RELOAD WORD <SelectCARA,CARB,COPC,CM>

       .WORD 000000   ; CURRENT ADDRESS REGISTER A SEG/TAG
       .WORD DESTNT   ; CURRENT ADDRESS REGISTER A OFFSET
                      ; <This local address is the destination>

       .WORD 101400   ; CURRENT ADDRESS REGISTER B SEG/TAG
       .WORD 000000   ; CURRENT ADDRESS REGISTER B OFFSET
                      ; <This global Q-Bus address is the source>
                      ; <This corresponds to address 600000 on
                      ; the Q-Bus>
                      ; <The DTC uses physical addresses only>

       .WORD 000013.  ; CURRENT OPERATION COUNT <Transfer 13 words>

       .WORD 000000   ; CHANNEL MODE REGISTER HIGH
       .WORD 000060   ; CHANNEL MODE REGISTER LOW
                      ; <No match conditions, do nothing upon
                      ; completion, transfer type = Demand
                      ; Dedicated w/Bus Hold,  CARB = source, word
                      ; transfers> <Notice how similar this reload
                      ; table is to the one in EXAM2. By utilizing
                      ; the flip bit in the CM Reg Low, no further
                      ; changes were necessary to use the table in
                      ; this example>

DESTNT:.BLKW   13.

       .END START
```

```
.TITLE EXAM4.MAC

; This program transfers data from global Q-Bus addresses to other
; global Q-Bus addresses.  This program should be compiled and
; linked on the development system, and then downloaded into the
; KXJ11-CA using the KXJ11-CA Software Toolkit. Once the program
; has been compiled and linked use the following commands to
; execute it and verify its success.
;
; <Use Q-Bus ODT to deposit values in locations
; 600000(8)-->600030(8).
; These values will be the source for this operation>
;
; @600000/000001   ! Deposit source values
;          .
;          .
; @600030/000001
; @P               ! Use the 'P' command to return to the system
                     prompt
;
; .KUI
; KUI>SET n        ! Where n is the appropriate KXJ11-CA
; KUI>LOAD EXAM4
; KUI>EXECUTE
; KUI>EXIT
;
; <Use Q-Bus ODT to examine the destination locations to verify
; that the operation was successful>
; @610000/xxxxxx
;         .
;         .
; @610030/xxxxxx
; @P               ! Return to system prompt

; SET UP REGISTER ASSIGNMENTS

MMREG   =    174470    ; MASTER MODE REGISTER
CMDREG  =    174454    ; COMMAND REGISTER
CASTF0  =    174446    ; CHANNEL 0 CHAIN ADDRESS SEGMENT/TAG FIELD
CAOF0   =    174442    ; CHANNEL 0 CHAIN ADDRESS OFFSET FIELD

START:  MOVB #130,MMREG      ; LOAD MASTER MODE REG TO DISABLE DTC

        CLRB CMDREG          ; RESET THE DTC

        MOV  #0,CASTF0       ; LOAD THE CHAIN ADDRESS REGISTER
                             ; SEG/TAG
        MOV  #RELOAD,CAOF0   ; LOAD THE CHAIN ADDRESS REGISTER
                             ; OFFSET

        MOVB #131,MMREG      ; LOAD MASTER MODE REG TO ENABLE DTC

        MOVB #102,CMDREG     ; SET SOFTWARE REQUEST CHANNEL 0

        MOVB #240,CMDREG     ; START CHAIN CHANNEL 0

        BR   .               ; STAY HERE WHILE THE USER VERIFIES
                             ; THAT THE PROGRAM WAS SUCCESSFUL
; CHAIN LOAD REGION

RELOAD: .WORD 001602    ; RELOAD WORD <Select CARA,CARB,COPC,CM>

        .WORD 101400    ; CURRENT ADDRESS REGISTER A SEG/TAG
        .WORD 000000    ; CURRENT ADDRESS REGISTER A OFFSET
                        ; <This global Q-Bus address is the source>
                        ; <This corresponds to Q-Bus address
                        ; 600000(8)>
        .WORD 101400    ; CURRENT ADDRESS REGISTER B SEG/TAG
        .WORD 010000    ; CURRENT ADDRESS REGISTER B OFFSET
                        ; <This global Q-Bus address is the
                        ; destination>
                        ; <This corresponds to Q-Bus address
                        ; 610000(8)>

        .WORD 000013.   ; CURRENT OPERATION COUNT <Transfer 13
                        ; words>

        .WORD 000000    ; CHANNEL MODE REGISTER HIGH
        .WORD 000040    ; CHANNEL MODE REGISTER LOW
                        ; <No match conditions, do nothing upon
                        ; completion, transfer type = Demand
                        ; Dedicated w/Bus Hold,
                        ; CARA = source, word transfers>

        .END START
```

```
.TITLE EXAM5.MAC

; This program demonstrates how chaining is implemented using the
; DTC.  A local to local transfer will be initiated under program
; control. Then, using the chaining feature of the DTC, a local to
; global transfer will be performed, followed by a global to
; global transfer, and finally a global to local transfer.  The
; following diagram illustrates these transfers.
;
;
;                KXJ11-CA Memory                  Q-Bus Memory
;
;Transfer #1    -------------                   ------------
;          +---|             |                  |            |
;          |   |             |                  |            |
;          |   |             |                  |            |
;          |   |-------------|  Transfer #2     |----------- |   Transfer #3
;          +-->|             |  --------------->|            |---+
;              |-------------|                  |----------- |   |
;              |             |                  |            |   |
;              |-------------|  Transfer #4     |----------- |   |
;              |             | <--------------- |            |<--+
;              |-------------|                  |----------- |
;
; This program should be compiled and linked on the development
; system and then downloaded into the KXJ11-CA using the KXJ11-CA
; Software Toolkit. Once the program has been compiled and linked
; use the following commands to execute it and verify its
; success.
;
; <Use Q-Bus ODT to clear the memory locations 600000(8) -->
; 600030(8) and 6100000(8) --> 610030(8) before executing the
; program>
;
; .KUI
; KUI>SET n          ! Where n is the appropriate KXJ11-CA
; KUI>LOAD EXAM5
; KUI>EXECUTE
; KUI>ODT            ! Use KUI ODT to verify that the destination
; ODT> .               contents are accurate
; ODT> .
; ODT>CTRL/C
; KUI>EXIT
;
; <Use Q-Bus ODT to examine the contents of the intermediate
; destinations to verify their accuracy>
; SET UP REGISTER ASSIGNMENTS

MMREG   =   174470  ; MASTER MODE REGISTER
CMDREG  =   174454  ; COMMAND REGISTER
CASTF0  =   174446  ; CHANNEL 0 CHAIN ADDRESS SEGMENT/TAG FIELD

CAOF0   =   174442  ; CHANNEL 0 CHAIN ADDRESS OFFSET FIELD
```

```
START:  MOVB #130,MMREG     ; LOAD MASTER MODE REG TO DISABLE DTC

        CLRB CMDREG         ; RESET THE DTC
        MOV  #0,CASTF0      ; LOAD THE CHAIN ADDRESS REGISTER
                            ; SEG/TAG
        MOV  #LOAD1,CAOF0   ; LOAD THE CHAIN ADDRESS REGISTER
                            ; OFFSET

        MOVB #131,MMREG     ; LOAD MASTER MODE REG TO ENABLE DTC

        MOVB #102,CMDREG    ; SET SOFTWARE REQUEST CHANNEL 0

        MOVB #240,CMDREG    ; START CHAIN CHANNEL 0

        BR   .              ; STAY HERE WHILE THE USER VERIFIES
                            ; THAT THE PROGRAM WAS SUCCESSFUL

; CHAIN LOAD REGION

LOAD1: .WORD 001603    ; RELOAD WORD <Select CARA,CARB,COPC,CM,CA>

        .WORD 000000   ; CURRENT ADDRESS REGISTER A SEG/TAG
        .WORD AREA1    ; CURRENT ADDRESS REGISTER A OFFSET
                       ; <This local address is the source of
                       ; transfer #1>

        .WORD 000000   ; CURRENT ADDRESS REGISTER B SEG/TAG
        .WORD AREA2    ; CURRENT ADDRESS REGISTER B OFFSET
                       ; <This local address is the destination of
                       ;  transfer #1>

        .WORD 000013.  ; CURRENT OPERATION COUNT <Transfer 13
                       ; words>

        .WORD 000000   ; CHANNEL MODE REGISTER HIGH
        .WORD 100040   ; CHANNEL MODE REGISTER LOW
                       ; <No match conditions, chain reload upon
                       ; completion, transfer type = Demand
                       ; Dedicated w/Bus Hold, CARA = source, word
                       ; transfers>
        .WORD 000000   ; CHAIN ADDRESS REGISTER SEG/TAG
        .WORD LOAD2    ; CHAIN ADDRESS REGISTER OFFSET
                       ; <This address points to the new chain
                       ; table>

LOAD2 : .WORD 001603   ; RELOAD WORD <Select CARA,CARB,COPC,CM,CA>

        .WORD 000000   ; CURRENT ADDRESS REGISTER A SEG/TAG
        .WORD AREA2    ; CURRENT ADDRESS REGISTER A OFFSET
                       ; <This local address is the source of
                       ; transfer #2>

        .WORD 101400   ; CURRENT ADDRESS REGISTER B SEG/TAG
        .WORD 000000   ; CURRENT ADDRESS REGISTER B OFFSET
                       ; <This global address is the destination of
                       ;  transfer #2 - 600000(8)>
```

```
.WORD 000013.           ; CURRENT OPERATION COUNT <Transfer 13
                        ; words>

        .WORD 000000    ; CHANNEL MODE REGISTER HIGH
        .WORD 100040    ; CHANNEL MODE REGISTER LOW
                        ; <No match conditions, chain reload upon
                        ; completion, transfer type = Demand
                        ; Dedicated w/Bus Hold, CARA = source, word
                        ; transfers>

        .WORD 000000    ; CHAIN ADDRESS REGISTER SEG/TAG
        .WORD LOAD3     ; CHAIN ADDRESS REGISTER OFFSET
                        ; <This address points to the new chain
                        ; table>

LOAD3 : .WORD 001603    ; RELOAD WORD <Select CARA,CARB,COPC,CM,CA>

        .WORD 101400    ; CURRENT ADDRESS REGISTER A SEG/TAG
        .WORD 000000    ; CURRENT ADDRESS REGISTER A OFFSET
                        ; <This global address is the source of
                        ; transfer #3>
                        ;  <600000(8)>

        .WORD 101400    ; CURRENT ADDRESS REGISTER B SEG/TAG
        .WORD 010000    ; CURRENT ADDRESS REGISTER B OFFSET
                        ; <This global address is the destination of
                        ; transfer #3 - 610000(8)>

        .WORD 000013.   ; CURRENT OPERATION COUNT <Transfer 13
                        ; words>
        .WORD 000000    ; CHANNEL MODE REGISTER HIGH
        .WORD 100040    ; CHANNEL MODE REGISTER LOW
                        ; <No match conditions, chain reload upon
                        ; completion, transfer type = Demand
                        ; Dedicated w/Bus Hold,  CARA = source, word
                        ; transfers>

        .WORD 000000    ; CHAIN ADDRESS REGISTER SEG/TAG
        .WORD LOAD4     ; CHAIN ADDRESS REGISTER OFFSET
                        ; <This address points to the new chain
                        ; table>


LOAD4 : .WORD 001602    ; RELOAD WORD <Select CARA,CARB,COPC,CM>

        .WORD 101400    ; CURRENT ADDRESS REGISTER A SEG/TAG
        .WORD 010000    ; CURRENT ADDRESS REGISTER A OFFSET
                        ; <This global address is the source of
                        ; transfer #4>
                        ;  <610000(8)>

        .WORD 000000    ; CURRENT ADDRESS REGISTER B SEG/TAG
        .WORD AREA3     ; CURRENT ADDRESS REGISTER B OFFSET
                        ; <This local address is the destination of
                        ;  transfer #4>
        .WORD 000013. ; CURRENT OPERATION COUNT <Transfer 13
                      ; words>

        .WORD 000000  ; CHANNEL MODE REGISTER HIGH
        .WORD 000040  ; CHANNEL MODE REGISTER LOW
                      ; <No match conditions, do nothing upon
                      ; completion, transfer type = Demand
                      ; Dedicated w/Bus Hold, CARA = source,
                      ; word transfers>

AREA1 : .WORD   1,2,3,4,5,6,7,6,5,4,3,2,1
AREA2 : .BLKW   13.
AREA3 : .BLKW   13.

        .END START
```

```
.TITLE  EXAM6.MAC

; This program demonstrates how to initiate a DTC operation from
; the arbiter CPU. This program will transfer a block of data from
; Q-Bus memory to KXJ11-CA memory.  All of the information
; necessary for the transfer will reside in Q-Bus memory (chain
; table, source data). This program should be compiled, linked,
; and run on the arbiter development system. After the program
; executes, use the following KUI commands to verify the transfer.
;
; .KUI
; KUI>SET n           ! Where n is the appropriate KXJ11-CA
; KUI>ODT
; ODT>5000/xxxxxx     ! Examine locations 5000 --> 5030 to verify
;       .               that the data was transferred correctly
;       .
; ODT>5030/xxxxxx
; ODT>CTRL/C
; KUI>EXIT
;
; Two-port RAM register definitions

TPR0=160100
TPR2=160104
TPR3=160106

.MCALL  .EXIT

START: MOV #100000,TPR3 ; Place Chain Address Reg Seg/Tag in TPR3
       MOV #LOAD,TPR2   ; Place Chain Address Reg Offset in TPR2

       BIS #2,TPR0      ; Issue DMA Load command to the command
                        ; register

.EXIT

LOAD   : .WORD  001602    ; RELOAD WORD <Select CARA,CARB,COPC,CM>
         .WORD  100000    ; CARA SEG/TAG <Select Q-Bus address as
                          ; source>
         .WORD  SOURCE    ; CARA OFFSET

         .WORD  000000    ; CARB SEG/TAG <Select KXT address 5000
                          ; as destination>

         .WORD  005000    ; CARB OFFSET

         .WORD  000013.   ; COPC <Op-count = 13 words>

         .WORD  000000    ; CM High
         .WORD  000040    ; CM Low <select no termination options,
                          ; software hog-mode, CARA = source, word
                          ; transfers>

SOURCE:  .WORD  1,2,3,4,5,6,7,6,5,4,3,2,1

         .END   START
```

CHAPTER 5
PARALLEL I/O CONTROLLER

5.1 OVERVIEW

The parallel I/O controller (PIO) is designed around the AmZ8036 chip. For details on the operation of the AmZ8036, refer to the Z8036 Counter/Timer and Parallel I/O Unit Technical Manual included in this documentation package. The information that follows summarizes and describes the PIO functions implemented on the KXJ11-CA.

The KXJ11-CA PIO has the following features.

- Two 8-bit, double buffered, bidirectional I/O ports
- A 4-bit special purpose I/O port
- Four handshake modes
- REQUEST signal for utilizing the DMA controller
- Pattern recognition logic
- Three independent 16-bit counter/timers

The two 8-bit ports (A and B) are identical except that Port B can provide external access to Counter/Timers 1 and 2. Each port may be configured under program control as a single or double-buffered port with handshake logic, or as a bit port for control applications. Pattern recognition logic is also included in each port. This logic allows interrupt generation whenever a specific pattern is recognized. Ports A and B may be linked to form a 16-bit port with handshake logic.

When Port A or B is used as a port with handshake logic, the control lines are supplied by a special 4-bit port (Port C). If no handshake lines are required, then Port C may be used as a bit port. Port C also provides external access to Counter/Timer 3 and a REQUEST line that allows the PIO to utilize the DMA controller when transferring data.

The PIO supplies three identical 16-bit counter/timers. These counter/timers operate at a frequency of 2 MHz, which provides a resolution of 500 ns. Each counter/timer may operate with one of three output duty cycles: pulse, one-shot, or square-wave. In addition, each unit may operate as retriggerable or non-retriggerable. Timers 1 and 2 can be cascaded to make a 32-bit timer. External count, external gate, and external trigger lines are provided for all three counter/timers.

5.2 PARALLEL I/O PORT (PIO) REGISTERS

The PIO is designed around the AmZ8036 chip and consists of two 8-bit ports: one 4-bit port and a counter/timer. Table 5-1 summarizes the registers associated with the PIO. All the registers in Table 5-1 reside in the AmZ8036 chip, with the exception of the I/O Buffer Control Register, which resides in the GAS on-board gate array (DC7037B). The sections that follow give brief descriptions of the PIO registers.

Table 5-1 PIO Registers

Master Control Registers			
Master Interrupt Control Register	17777000		
Master Configuration Control Register	17777002		
Port Specification Registers	A	B	
Port Mode Specification Register	17777100	17777120	
Port Handshake Specification Register	17777102	17777122	
Port Command and Status Register	17777020	17777022	
Bit Path Definition Registers	A	B	C
Data Path Polarity Registers	17777104	17777124	17777012
Data Direction Registers	17777106	17777126	17777014
Special I/O Control Registers	17777110	17777130	17777016
Pattern Definition Registers	A	B	
Pattern Polarity Registers (PPR)	17777112	17777132	
Pattern Transition Registers (PTR)	17777114	17777134	
Pattern Mask Register (PMR)	17777116	17777136	
Port Data Registers	A	B	C
	17777032	17777034	17777036
PIO Counter/Timer Control Registers	C/T 1	C/T 2	C/T 3
PIO Counter/Timer Mode Specification	17777070	17777072	17777074
PIO Counter/Timer Command and Status	17777024	17777026	17777030
PIO Counter/Timer Time Constant (MSB)	17777054	17777060	17777064
PIO Counter/Timer Time Constant (LSB)	17777056	17777062	17777066
PIO Counter/Timer Current Count (MSB)	17777040	17777044	17777050
PIO Counter/Timer Current Count (LSB)	17777042	17777046	17777052
Interrupt Related Registers	A	B	C/T
Interrupt Vector Register	17777004	17777006	17777010
Current Vector Register	17777076		
I/O Buffer Control Register	17777140		

5.2.1 Master Control Registers

The Master Control Registers affect the overall operation of the PIO. There are two Master Control Registers: the Master Interrupt Control Register and the Master Configuration Control Register. All bits of these two registers are cleared upon hardware reset, except bit 0 of the Master Interrupt Control Register, which is set. Both registers are read/write.

5.2.1.1 Master Interrupt Control Register -- See Figure 5-1.

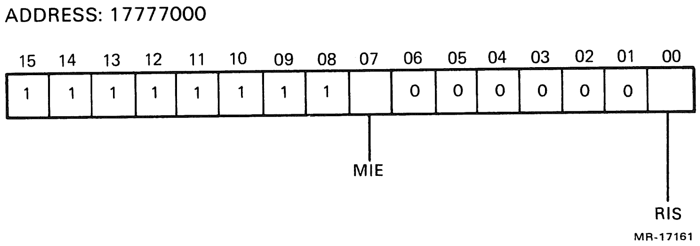


Figure 5-1 Master Interrupt Control Register

Bits	Name	Description
15:8		Not used (read as ones)
7	MIE	Master interrupt enable - When cleared, prevents the PIO from requesting interrupt service or responding to an interrupt acknowledge cycle. When set, enables interrupts.
6:1		Must be zero
0	RIS	Reset - Set upon hardware reset. Must be explicitly cleared. When set, reads of other PIO registers will yield a value of one, and writes will be ignored.

5.2.1.2 Master Configuration Control Register -- See Figure 5-2.

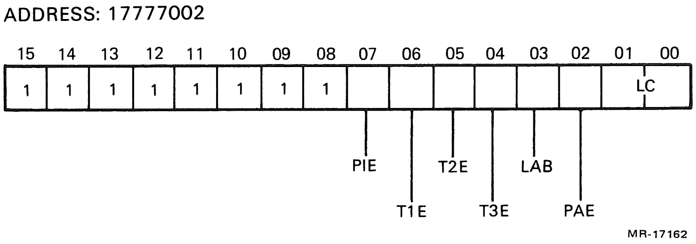


Figure 5-2 Master Configuration Control Register

Bits	Name	Description
15:8		Not used (read as ones)
7	PBE	Port B enable - When cleared, inhibits port B from issuing an interrupt request, and forces the port B I/O lines into a high impedance state.
6	T1E	Counter/timer 1 enable - When cleared, inhibits counter/timer 1 from issuing an interrupt request, and clears bit 0 of the Counter/Timer 1 Command and Status Register.
5	T2E	Counter/timer 2 enable - When cleared, inhibits counter/timer 2 from issuing an interrupt request, and clears bit 0 of the Counter/Timer 2 Command and Status Register.
4	T3E	Port C and counter/timer 3 enable - When cleared, inhibits port C and counter/timer 3 from issuing an interrupt request. Also clears bit 0 of the Counter/Timer 3 Command and Status Register, and forces the port C I/O lines into a high impedance state.
3	LAB	Port link control - When cleared, allows port A and port B to operate independently. When set, links ports A and B to form a 16-bit port. When the ports are linked, only port A's Handshake and Command and Status Registers are used. Port B is specified as a bit port and its pattern matching capability is disabled. When linked, port B must be read or written before port A. If the ports are to be linked, this bit must be set before the ports are enabled.

Bits	Name	Description
2	PAE	Port A enable - When cleared, inhibits port A from issuing an interrupt request and forces the port A I/O lines into a high impedance state.
1:0	LC	Counter/timer link control - Specifies if and how counter/timers 1 and 2 are linked. The counter/timers must be linked before they are enabled.

LC	Configuration
00	Counter/timers are independent
01	C/T 1s output (inverted) enables C/T 2
10	C/T 1s output (inverted) triggers C/T 2
11	C/T 1s output (inverted) is C/T 2s count input

5.2.2 Port Specification Registers

The Port Specification Registers define the operating characteristics of ports A and B. There are three types of Port Specification Registers: Mode, Handshake, and Command and Status. Each port (A and B) has one set of these three registers.

5.2.2.1 Port Mode Specification Registers (Ports A and B) -- These registers are read/write (Figure 5-3). They are cleared during hardware reset.

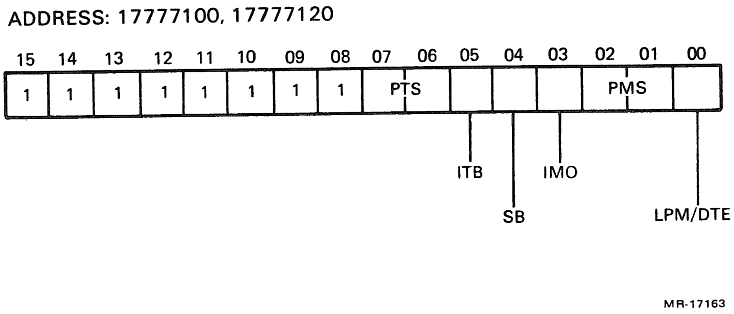


Figure 5-3 Port Mode Specification Registers (Ports A and B)

Bits	Name	Description
15:8		Not used (read as ones)
7:6	PTS	Port type select - Specifies the port type.

PTS	Port Type
00	Bit port (no handshake)
01	Input port with handshake
10	Output port with handshake
11	Reserved

Bits	Name	Description
5	ITB	Interrupt on two bytes - When cleared, the Interrupt Pending (IP) bit for this port (bit 5 of the Port Command and Status Register) is set when one byte of data is available for transfer. When this bit is set, IP is set when two bytes of data are available for transfer. For an input port, IP is set when the Input Data Register is full. For an output port, IP is set when the Output Data Register is empty. This bit must be cleared for ports specified as bit ports, single- buffered ports, or bidirectional ports.
4	SB	Single buffered mode - When cleared, specifies that this port is double-buffered. When set, specifies that this port is single-buffered. This bit is always cleared for bit ports.
3	IMO	Interrupt on match only - When set, an interrupt is generated when data moved into the Input Data Register or out of the Output Data Register matches the pattern specification.
2:1	PMS	Pattern mode specification - Defines the operation of the pattern match logic.

PMS	Pattern Mode
00	Disable pattern matching
01	AND mode
10	OR mode
11	OR-priority encoded vector mode

Bits	Name	Description
0	LPM/DTE	Latch on pattern match (LPM) or deskew timer enable (DTE) - This is a dual function bit. The LPM function is active when the port is used as a bit port. The DTE function is active when the port is specified as an output port with handshake logic. If LPM is set, the port latches input data when a pattern match is detected. If LPM is cleared, pattern matches are detected but the data read from the port is the current (unlatched) value. If DTE is set, the deskew timer is active and can perform delay functions (see the description of the Port Handshake Specification Register that follows this section). When DTE is cleared, the deskew timer is not active.

5.2.2.2 Port Handshake Specification Registers (Ports A and B) -- The Port Handshake Specification Registers (Figure 5-4) determine the parameters of a handshake operation. A Port Handshake Specification Register is ignored if a port is configured as a bit port. These registers are cleared during reset. Access is read/write.

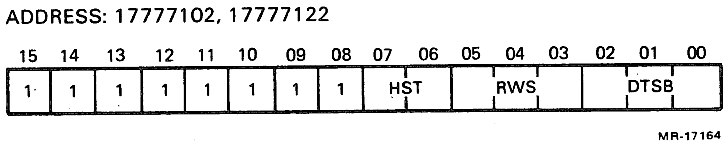


Figure 5-4 Port Handshake Specification Registers (Ports A and B)

Bits	Name	Description
15:8		Not used (read as ones)
7:6	HST	Handshake type - Specify the type of handshake operation performed.

HST	Handshake Type
00	Interlocked
01	Strobed
10	Pulsed
11	3-Wire

The pulsed and 3-wire handshake must not be specified for bidirectional ports. Only one port at a time can use the pulsed handshake. If one port uses the 3-wire handshake, the other port must be a bit port.

Bits Name Description

5:3 RWS

Request/Wait - Defines how this port implements the request function. The wait function is not implemented on the KXJ11-CA.

RWS	Request Function
000	Request disabled
001	Reserved
010	Reserved
011	Reserved
100	Special request
101	Output request
110	Reserved
111	Input request

Only port A can participate in a request. Port B must be programmed as a bit port.

2:0 DTSB

Deskew time specification - Defines the minimum number of PIO clock cycles of delay between the time a new byte of data is output, and the time the handshake mechanism indicates that new data is available. The PIO clock has a period of 250 ns. A deskew time of zero is defined by setting DTE to zero in the Port Mode Specification Register.

DTSB	Deskew Clock Cycles
000	2
001	4
010	6
011	8
100	10
101	12
110	14
111	16

5.2.2.3 Port Command and Status Registers (Ports A and B) -- See Figure 5-5.

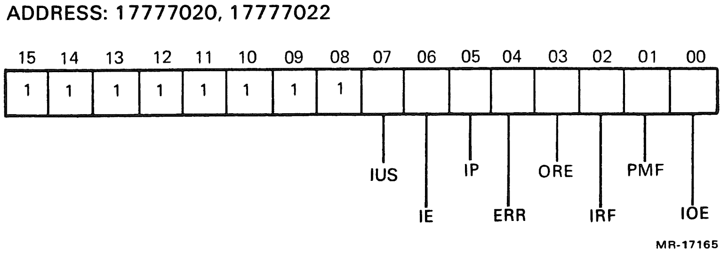


Figure 5-5 Port Command and Status Registers (Ports A and B)

Bits	Name	Description
15:8		Not used (read as ones)
7	IUS	Interrupt under service - When set, indicates that this port is engaged in an interrupt acknowledge sequence. Interrupt requests at the same level or lower are disabled. This bit is read/write and is cleared during reset.
6	IE	Interrupt enable - When cleared, this port is prevented from requesting an interrupt or engaging in an interrupt acknowledge sequence. When set, these interrupts are enabled. The bit is read/write and is cleared during reset.
5	IP	Interrupt pending - When set, indicates that this port requires service because of a pattern match, a handshake operation, or an error. When cleared, indicates that the port does not require service. This bit is read/write and is cleared during reset. IUS, IE, and IP are written according to the following command codes. **Bits <7:5> Command** 000 Null (no effect) 001 Clear IP and IUS 010 Set IUS 011 Clear IUS 100 Set IP 101 Clear IP 110 Set IE 111 Clear IE
4	ERR	Interrupt error - This bit is meaningful only if this port has been configured as a bit port, and pattern matching has been enabled. When set, indicates that a pattern match occurred before a previous match could be acknowledged. This bit is read-only (writes to it are ignored), and is cleared during reset.
3	ORE	Output register empty - When set, indicates that this output port's Output Data Register is empty. Can be cleared only by writing to the Output Data Register. This bit is read-only (writes to it are ignored) and is set during reset.

Bits Name Description

2 IRF Input register full - When set, indicates that this input port's Input Data Register is full. Can be cleared only by reading the Input Data Register. This bit is read-only (writes to it are ignored), and is cleared upon reset.

1 PMF Pattern match flag - When set, this bit set indicates the occurrence of a pattern match, if pattern matching is enable for this port. This bit is read-only (writes to it are ignored), and is cleared upon reset.

0 IOE Interrupt on error - This bit is meaningful only for bit ports with pattern matching enabled. When cleared, this bit prevents an interrupt from being issued by this port if an error occurs in pattern matching. When set, this bit allows these interrupts. The bit is ignored by ports with handshake and should be cleared for these ports. The bit is read/write.

5.2.3 Bit Path Definition Registers

Each port (A, B, and C) has one set of Bit Path Definition Registers. They include the Data Path Polarity, Data Direction, and Special I/O Control Registers. Only the four least significant bits of the registers are valid for the Port C registers.

5.2.3.1 Data Path Polarity Registers -- The Data Path Polarity Registers (Figure 5-6) define whether the bits in a port are inverting or non-inverting. These registers are cleared during reset. Access is read/write.

ADDRESS: 17777104, 17777124, 17777012

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	POLARITY							

MR-17212

Figure 5-6 Data Path Polarity Registers (Ports A, B, and C)

Bits	Name	Description
15:8		Not used (read as ones)
7:0	DPP	Data path polarity - If a bit is set, the corresponding bit path for this port is inverting (asserted LOW). If a bit is cleared, the corresponding bit path for this port is non-inverting (asserted HIGH).

5.2.3.2 Data Direction Registers -- The Data Direction Registers (Figure 5-7) define the data direction of each bit in a port. These registers are ignored by ports with handshake logic and are cleared upon reset. Access is read/write.

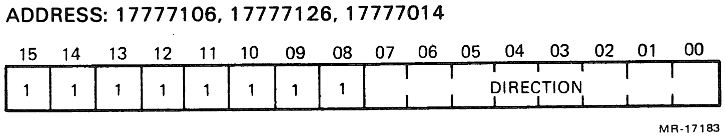


Figure 5-7 Data Direction Registers (Ports A, B, and C)

Bits	Name	Description
15:8		Not used (read as ones)
7:0	DD	Data direction - If a bit is set, the corresponding bit of this port is specified as an input bit. If a bit is cleared, the corresponding bit of this port is defined as an output bit.

5.2.3.3 Special I/O Control Registers -- The Special I/O Registers (Figure 5-8) allow special characteristics to be defined for a port's data path. These registers are cleared during reset. Access is read/write.

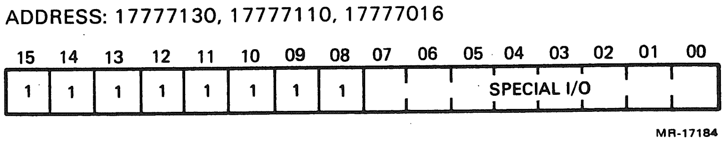


Figure 5-8 Special I/O Registers (Ports A, B, and C)

Bits	Name	Description
15:8		Not used (read as ones)
7:0	SIO	Special input/output - If a bit is set, the corresponding bit of this port is specified as a 1's catcher for input. A 1's catcher functions by automatically latching a 1 if the input goes to 1. The 1's catcher is cleared only by writing a zero to the Input Data Register.

5.2.4 Pattern Definition Registers

The Pattern Definition Registers (Figures 5-9 through 5-11) are used collectively to specify a match pattern for each bit in Port A or Port B. The pattern specification for any bit (x) is summarized in Table 5-2. These registers are cleared during reset. Access is read/write.

Table 5-2 Pattern Specifications

PPRx	PTRx	PMRx	Bit x Match Condition
0	0	0	Bit masked off
0	0	1	Bit masked off
0	1	0	Any transition
0	1	1	Any transition
1	0	0	Zero
1	0	1	One
1	1	0	One to zero transition
1	1	1	Zero to one transition

5.2.4.1 Pattern Polarity Registers (PPR) -- See Figure 5-9.

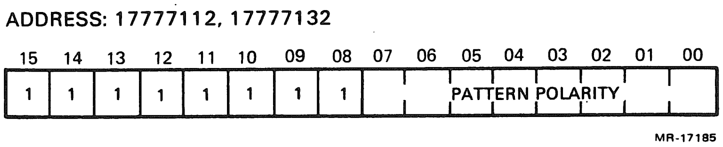


Figure 5-9 Pattern Polarity Registers (Ports A and B)

5.2.4.2 Pattern Transition Registers (PTR) -- See Figure 5-10.

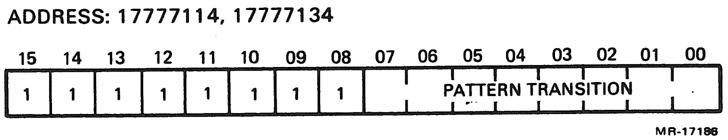


Figure 5-10 Pattern Transition Registers (Ports A and B)

5.2.4.3 Pattern Mask Registers (PMR) -- See Figure 5-11.

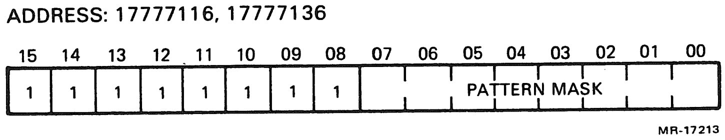


Figure 5-11 Pattern Mask Registers (Ports A and B)

5.2.5 Port Data Registers

Port Data registers are used to hold data that is read from or written to the PIO. The Port Data Register format for ports A and B is shown in Figure 5-12. The format for the Port C Data Register is shown in Figure 5-13. These registers are read/write and are unaffected by a reset.

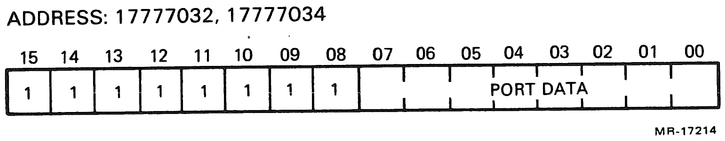


Figure 5-12 Port Data Registers (Ports A and B)

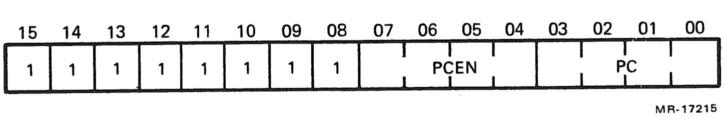


Figure 5-13 Port C Data Register

Bits	Name	Description
15:8		Not used (read as ones)
7:4	PCEN	Port C bit enable - used as a write-protect bit mask for bits <3:0>. A set bit in PCEN inhibits the writing of the corresponding bit in bits <3:0>. A cleared bit in PCEN enables the writing of the corresponding bit in bits <3:0>.
3:0	PC	Port C data - contains the four bits of data to be read by or written to Port C. Subject to masking according to the value of PCEN.

5.2.6 PIO Counter/Timer Control Registers

There are three PIO counter/timers numbered 1, 2, and 3. Each PIO counter/timer has a set of six control registers that specify the operation that the counter/timer performs. The registers are described in the paragraphs that follow.

5.2.6.1 PIO Counter/Timer Mode Specification -- Each counter/timer has a Mode Specification Register (Figure 5-14). These registers define an operational mode for a counter/timer, and specify which external control and status lines are used. They are cleared during reset. Access is read/write.

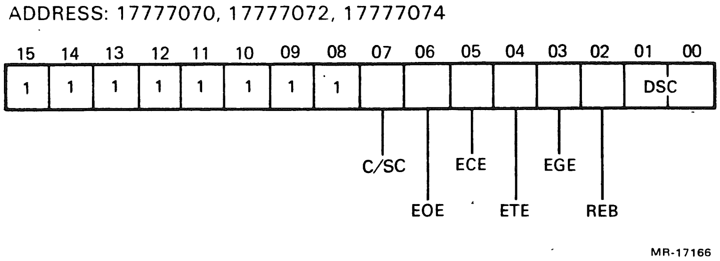


Figure 5-14 Counter/Timer Mode Specification (Counter/Timers 1, 2, and 3)

Bits	Name	Description
15:8		Not used (read as ones)
7	C/SC	Continuous/single cycle - When set, the time constant value initially used is reloaded, and the countdown sequence is repeated when the counter reaches zero. When cleared, the countdown sequence is terminated when the counter reaches zero.

Bits	Name	Description
6	EOE	External output enable - When set, the output of the counter/timer is provided on the I/O line associated with that particular counter/timer (see Table 5-3). This bit must be programmed as an output bit in the Data Direction Register of its port. When cleared, external access to the counter/timer is disabled.
5	ECE	External count enable - When set, the I/O line of the port associated with the counter/timer is used as an external counter input (see Table 5-3). The corresponding bit must be programmed as input. When cleared, external access is disabled.
4	ETE	External trigger enable - When set, the I/O line of the port associated with the counter/timer is used as a trigger input bit to the counter/timer (see Table 5-3). The corresponding bit must be programmed as input. When cleared, external access is disabled.
3	EGE	External gate enable - When set, the I/O line associated with the counter/timer is used as an external gate to the counter/timer (see Table 5-3). This allows the external line to suspend or continue the countdown in progress by toggling the line. When cleared, external access is disabled.
2	REB	Retrigger enable bit - When set, triggers that occur during a countdown sequence cause a new countdown to begin. When cleared, triggers that occur during a countdown sequence are ignored.
1:0	DSC	Output duty cycle select -

DSC	**Output Duty Cycle**
00	Pulse output
01	One-shot output
10	Square wave output
11	Reserved

External access to the counter/timers is provided through the following I/O lines (Table 5-3).

Table 5-3 Counter/Timer External Access

Function	C/T 1	C/T 2	C/T 3
Counter/Timer Output	Port B 4	Port B 0	Port C 0
Counter Input	Port B 5	Port B 1	Port C 1
Trigger Input	Port B 6	Port B 2	Port C 2
Gate Output	Port B 7	Port B 3	Port C 3

5.2.6.2 PIO Counter/Timer Command and Status -- . Each counter/timer has a Command and Status Register (Figure 5-15) that is used to control and monitor timer operation. These registers are cleared during reset.

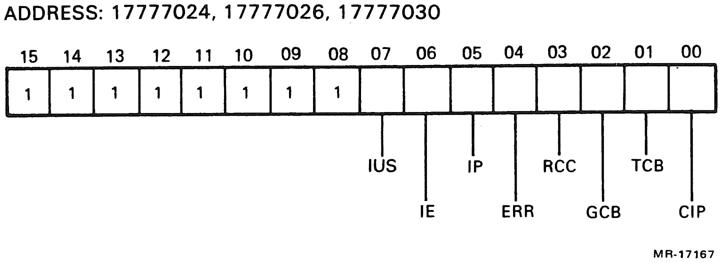


Figure 5-15 Counter/Timer Command and Status (Counter/Timers 1, 2, and 3)

Bits	Name	Description
15:8		Not used (read as ones)
7	IUS	Interrupt under service - When set, indicates that this counter/timer is engaged in an interrupt acknowledge sequence. Interrupt requests at the same level or lower are disabled. This bit is read/write.
6	IE	Interrupt enable - When cleared, this counter/timer is prevented from requesting an interrupt or engaging in an interrupt acknowledge sequence. When set, the interrupts are enabled. The bit is read/write.

Bits	Name	Description
5	IP	Interrupt pending - When set, indicates that this counter/timer requires service. The bit is automatically set each time the counter/timer reaches its terminal count. When cleared, indicates that the counter/timer does not require service. This bit is read/write.

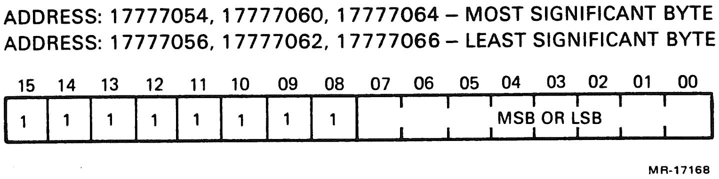
IUS, IE, and IP are written according to the following command codes.

Bits <7:5>	**Command**
000	Null (no effect)
001	Clear IP and IUS
010	Set IUS
011	Clear IUS
100	Set IP
101	Clear IP
110	Set IE
111	Clear IE

4	ERR	Interrupt error - When set, indicates that the counter/timer has reached a terminal count before the previous terminal count has been serviced. This bit is read-only.
3	RCC	Read counter control - When set, causes the contents of the Counter/Timer Current Count Register (which normally follows the down counter) to be frozen until the least significant byte of the register is read. This bit cannot be set unless the counter/timer is enabled in the Master Configuration Control Register.
2	GCB	Gate command bit - When set, starts or resumes the countdown sequence. When cleared, halts the countdown sequence. This bit is read/write.
1	TCB	Trigger command bit - When set, the down-counter is loaded with the time constant value, and a countdown sequence is initiated. This bit is write-only and is always read as zero.
0	CIP	Count in progress - When set, indicates that a countdown sequence is in progress. It is automatically set when the down-counter is loaded with the time constant value. It is automatically cleared when the down-counter reaches zero. This bit is read-only.

5.2.6.3 PIO Counter/Timer Time Constant -- Each counter/timer has a register that contains a time constant value. This value is loaded into the down-counter of a counter/timer when a trigger is detected. Each register is 16 bits wide and is accessed as two consecutive bytes (bit 7 of the MSB is bit 15 of the PIO Counter/Timer Time Constant Register). Refer to Figure 5-16 for the register format. These registers are read/write and are unaffected by a reset.

5.2.6.4 PIO Counter/Timer Current Count -- Each counter/timer has a Current Count Register (Figure 5-17). This register follows the contents of the appropriate down-counter until a 1 is written into the RCC bit of the Status/Control Register. When this happens, the contents of the Current Count Register are frozen until the least significant byte of the register is read. Then the register follows the contents of the down-counter again. The countdown sequence is not affected. Each register is 16 bits wide and is accessed as two consecutive bytes (bit 7 of the MSB is bit 15 of the Current Count Register). A reset forces the Current Count Register to follow the down-counter. Writes to the Current Count Register are ignored.

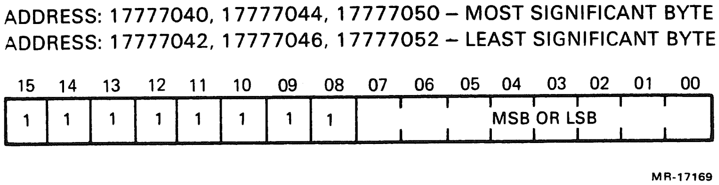


ADDRESS: 17777054, 17777060, 17777064 – MOST SIGNIFICANT BYTE
ADDRESS: 17777056, 17777062, 17777066 – LEAST SIGNIFICANT BYTE

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	MSB OR LSB							

MR-17168

Figure 5-16 Counter/Timer Time Constant (Counter/Timers 1, 2, and 3)



ADDRESS: 17777040, 17777044, 17777050 – MOST SIGNIFICANT BYTE
ADDRESS: 17777042, 17777046, 17777052 – LEAST SIGNIFICANT BYTE

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	MSB OR LSB							

MR-17169

Figure 5-17 Counter/Timer Current Count (Counter/Timers 1, 2, and 3)

5.2.7 Interrupt Related Registers

Interrupt related registers are registers used in the handling of PIO interrupts. Three of these are vector registers: one for Port A, one for Port B, and one shared by the three counter/timers. Another register is provided to indicate which devices need service in a polled environment.

5.2.7.1 Interrupt Vector Register -- The Interrupt Vector Register holds the vector used during an interrupt acknowledge operation. The native firmware initializes the vector for Port A at 200 (octal), the vector for Port B at 204, and the vector for the counter/timers at 210. If the MIE bit of the Master Interrupt Control Register is set, bits 1, 2, and 3 of the vector are affected as shown.

Ports A and B

OR-Priority Encoded Vector Mode:

Bit 3	Bit 2	Bit 1	Encodes the number of the highest
x	x	x	priority bit with a match.

All other modes (see Port Command and Status Register description):

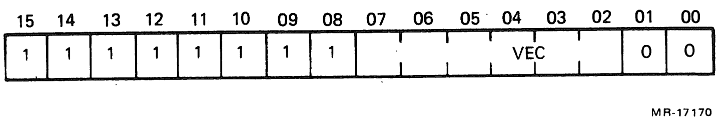
Bit 3	Bit 2	Bit 1	
ORE	IRF	PMF	No error
0	0	0	Error

Counter/Timers

Bit 2	Bit 1	
0	0	Counter/timer 3
0	1	Counter/timer 2
1	0	Counter/timer 1
1	1	Error

This register is read/write and is unaffected by a reset. The format of the Interrupt Vector Register is shown in Figure 5-18.

ADDRESS: 17777004, 17777006, 17777010



15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	VEC						0	0

Figure 5-18 Interrupt Vector Register

5.2.7.2 **Current Vector Register** -- The Current Vector Register is a read-only register. When read, it returns the vector that would have been returned during an interrupt acknowledge cycle, if the device had been assigned the highest priority interrupt pending. The order of priority (highest to lowest) is counter/timer 3, Port A, counter/timer 2, Port B, counter/timer 1. If no enabled interrupts are pending, or if the PIO is reset, the register will contain a pattern of all 1's. This is useful in a polled environment.

The format of the Current Vector Register is shown in Figure 5-19.

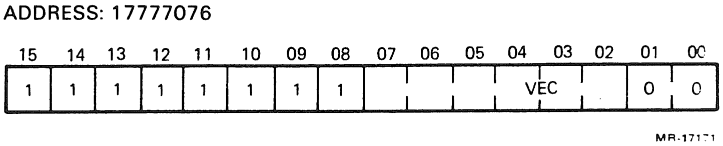


Figure 5-19 Current Vector Register

5.2.8 I/O Buffer Control Register

The PIO is protected from the connector by a set of IEEE 488 compatible buffers. The buffers are controlled by the I/O Buffer Control Register (Figure 5-20). The register allows the user to configure ports as inputs or outputs. Also, port driver buffers can be configured to operate in open collector or active pull-up mode. This register is cleared during reset. This is a write-only register.

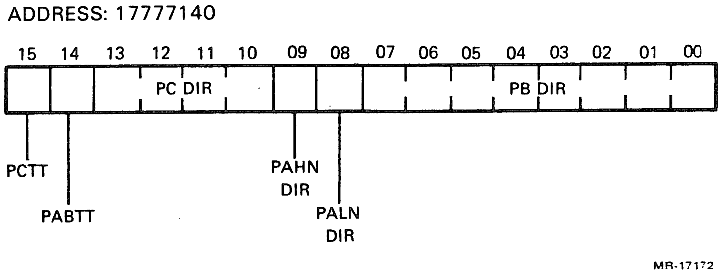


Figure 5-20 I/O Buffer Control Register

Bits	Name	Description
15	PCTT	Port C buffer control - When set, configures the Port C drivers as active pull-up drivers. When cleared, the Port C drivers are open collector drivers.
14	PABTT	Ports A and B buffer control - When set, configures the port A and B drivers as active pull-up drivers. When cleared, the port A and B drivers are open collector drivers.

Bits	Name	Description
13:10	PC DIR	Port C direction - If a bit is set, the corresponding Port C bit is a driver. If a bit is cleared, the corresponding Port C bit is a receiver.
9	PAHN DIR	Port A high nibble direction - When clear, the Port A high nibble bits <7:4> are receivers. When set, the Port A high nibble bits are drivers.
8	PALN DIR	Port A low nibble direction - When clear, the Port A low nibble bits <3:0> are receivers. When set, the Port A low nibble bits are drivers.
7:0	PB DIR	Port B direction - If a bit is set, the corresponding Port B bit is a driver. If a bit is cleared, the corresponding Port B bit is a receiver.

5.3 PROGRAMMING THE I/O PORTS

This section describes how to program the I/O ports and provide example programs. In particular this section describes how to use the I/O ports as bit ports, as ports with handshake logic, in 16-bit linked mode, and with the DMA controller. The use of the pattern recognition logic will also be discussed.

5.3.1 Programming the I/O Ports as Bit Ports

Using the I/O ports as bit ports provides up to 20 lines for control and status. Each bit in ports B and C may be independently configured as an input or output bit. Port A must be configured on a nibble (4-bit) basis.

Programming the PIO as a bit port is straightforward. First, the Port Mode Specification Register is used to select the port as a bit port with or without pattern matching. Then, the Bit Path Definition Registers are used to determine the polarity, direction, and special characteristics of the bits of the port. If pattern recognition is enabled, the Pattern Definition Registers must also be initialized. It is then a simple matter to write to the output data buffer to provide the correct control signals, and to read the input data buffer to monitor status.

The following program provides an example for using the PIO in the bit mode.

```
.TITLE  PIO1.MAC
;+
;  This program provides an example of how to program the PIO's
;  I/O ports as bit ports.  This program utilizes the PIO
;  loopback connector (Part #H3021 or 54-16227) which makes the
;  following connections:
;
;                      A0  --  B0
;                      A1  --  B1
;                           .
;                           .
;                      A7  --  B7
;                      C0  --  C3
;                      C1  --  C2
;
;  After this program has been assembled and linked on the
;  development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the program into the KXJ11-CA. The
;  program will execute as shown in the following example.
;
;  SET 2
;  LOAD PIO1.SAV
;  EXECUTE
;  !ODT
;  !
;  !001152
;  !R2/000000
;  !1154/041101
;  !001156/042103
;  !001160/043105
;  !001162/177507
;  !001164/041101
;  !001166/042103
;  !001170/043105
;  !001172/000107
;  !001174/000000
;  !^C
;  EXIT
;
;  A non-zero result in R2 indicates that an error has occurred.
;  (Try running the test without the loopback connector).
;  Location 1154 is the beginning of the output buffer. Location
;  1164 is the beginning of the input buffer.
;-
; Register Assignments

    MIC     ==  177000
    MCC     ==  177002
    PAMODE  ==  177100
    PAPOL   ==  177104
    PADDIR  ==  177106
    PASIO   ==  177110
    PADATA  ==  177032
```

```
        PBMODE   ==  177120
        PBPOL    ==  177124
        PBDDIR   ==  177126
        PBSIO    ==  177130
        PBDATA   ==  177034

        IOCNTL   ==  177140

START::
        MTPS     #340             ; Inhibit recognition of
                                  ; interrupts

        ;  Initialize PIO
        MOVB     #1,MIC           ; Reset device and inhibit interrupt
                                  ; requests
        CLRB     MIC              ; Enable device (interrupts still
                                  ; inhibited)

        ;  Set-up Port A
        CLRB     PAMODE           ; Port A:  bit port, no pattern match
        CLRB     PAPOL            ; Port A bits are non-inverting
        CLRB     PADDIR           ; Port A bits are output bits
        CLRB     PASIO            ; Normal output

        ;  Set-up Port B
        CLRB     PBMODE           ; Port B:  bit port, no pattern match
        CLRB     PBPOL            ; Port B bits are non-inverting
        MOVB     #377,PBDDIR      ; Port B bits are input bits
        CLRB     PBSIO            ; Normal input

        ;  Set-up the PIO buffers
        MOV      #1400,IOCNTL     ; configure the PIO buffers for
                                  ; A=output and B=input

        ;  Initialize GPRs
        MOV      #OUTBUF,R0       ; Point to data to be output
        MOV      #INBUF,R1        ; Point to input data buffer
        CLR      R2               ; R2 will indicate error status

        ;  Flush input buffer
        TSTB     PBDATA

        ;  Enable Ports A and B and send the data
        MOVB     #204,MCC         ; Enable ports A and B
```

```
1$:  MOVB    (R0)+,PADATA   ; Move data out of Port A
     NOP                    ;  .
                            ;  .
     MOVB    PBDATA,(R1)+   ; and into Port B

                            ; Test to see if done
     TSTB    (R0)           ; IF (R0) is positive
     BPL     1$             ; THEN transfer another byte
                            ; ELSE check if data is valid

     ;  Compare original data with received data
     MOV     #OUTBUF,R0     ; Point to output data buffer
     MOV     #INBUF,R1      ; Point to input data buffer
                            ; Test to see if done
2$:  TSTB    (R0)           ; IF (R0) is negative
     BMI     3$             ; THEN done comparing
                            ; ELSE do another compare
     CMPB    (R0)+,(R1)+    ; Compare bytes
     BEQ     2$             ; IF bytes are equal
                            ; THEN test another pair
                            ; ELSE indicate error
     INC     R2             ; A non-zero value of R2 indicates
                            ; an error
3$:  BR      .              ; Branch here upon completion

OUTBUF: .BYTE 101,102,103,104,105,106,107,-1
        .EVEN
INBUF:  .BLKB   7

.END    START
```

5.3.2 Programming the I/O Ports as Ports with Handshake

Ports A and B may be configured as ports with handshake to facilitate transferring data on a byte-by-byte basis. Port C is used to provide the handshake lines. In addition, Port C may use the REQUEST line to utilize a DMA controller to transfer the data. See Table 5-4 for a description of the Port C handshake lines. Figure 5-21 shows how two PIOs can be connected together to transfer data, and the handshake lines that are utilized.

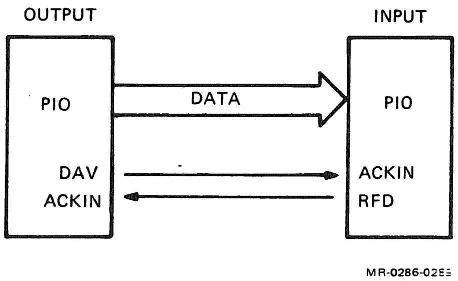


Figure 5-21   PIO Handshake Lines

Table 5-4   Port C Handshake Lines Port C Bits

Port A/B Configuration	Pin C3	Pin C2	Pin Cl	Pin C0
Ports A & B = Bit Ports	Bit I/O	Bit I/O	Bit I/O	Bit I/O
Port A = Input or Output (Interlocked, Strobed, or Pulsed Handshake)*	RFD or DAV	ACKIN	REQUEST or Bit I/O	Bit I/O
Port B = Input or Output (Interlocked, Strobed, or Pulsed Handshake)*	REQUEST or Bit I/O	Bit I/O	RFD or DAV	ACKIN
Port A or B = Input Port (3-Wire Handshake)	RFD (Output)	DAV (Input)	REQUEST or Bit I/O	DAC (Output)
Port A or B = Output Port (3-Wire Handshake)	DAV (Output)	DAC (Input)	REQUEST or Bit I/O	RFD (Input)
Port A or B = Bidirectional (Interlocked or Strobed Handshake)	Not implemented on the KXJ11-CA			

* Ports A and B may be specified at the same time as input or output ports with the Interlocked, Strobed, or Pulsed Handshakes, if neither port uses REQUEST. Only one port can use the Pulsed Handshake at a time.

The handshakes that are available are Interlocked, Strobed, Pulsed, and 3-Wire. A short description of each handshake type follows.

When using the Interlocked Handshake, any action by the PIO must be acknowledged by the external device before the next action can take place. In other words, an output port does not indicate that it has new data available until the external device indicates that it is ready for data. Likewise, an input port does not indicate that it is ready for new data until the external device indicates that the previous byte of data is no longer available (thereby acknowledging the input port's acceptance of the last byte).

The Strobed Handshake uses external logic to "strobe" data into or out of a port. In contrast to the Interlocked handshake, the signal indicating that the port is ready for another data transfer operates independently of the ACKIN input. External logic must ensure that data transfers at the appropriate speed.

The Pulsed Handshake is used for mechanical devices that require data to be held for relatively long periods of time, in order to be gated in or out of the device. The logic is similar to the Interlocked Handshake, except that Counter/Timer 3 is linked to the handshake logic to add the appropriate delays to the handshake lines.

The 3-Wire Handshake may be used so one output port can communicate to several input ports simultaneously. This is essentially the same as the Interlocked Handshake, except that two individual lines are used to indicate when an input port is ready for data (RFD), and when the input port has accepted data (DAC). Because this handshake requires three lines, only one port can use the 3-Wire Handshake at a time.

When Ports A and B are configured as ports with handshake logic, they must also be configured as single or double-buffered. Double-buffering a port allows more time for the interrupt service routine to respond to a data transfer. A second byte of data is input to or output from the port before the interrupt for the first byte is serviced. A single-buffered port is used where it is important to have byte-by-byte control over the transfer, or where it is important to enter the interrupt service or routine in a fixed amount of time after the data has been accepted or output.

The REQUEST line may also be used by ports with handshake. This control line enables the PIO to signal the DMA controller that the port wishes to transfer data without CPU intervention. The operation of the REQUEST line depends upon the Interrupt on Two Bytes (ITB) bit in the Port Mode Specification Register. If ITB = 0, then the REQUEST line is asserted anytime a byte is available for transfer. If ITB = 1, then the REQUEST line is not asserted until two bytes are available for transfer. The implementation of the PIO on the KXJ11-CA requires that only one port be used for DMA transfers. Since the REQUEST line utilizes one of the Port C bits, the other port must be programmed as a bit port. Also, bit 1 in the Port C Polarity Register must be set to insure the correct polarity for the REQUEST line if the DTC is used with the PIO.

When using the PIO with the DTC, address 17777033 must be used for the Port A Data Register, and address 17777035 must be used for the Port B Data Register.

The following examples display the capabilities of the PIO used as a port with handshake.

```
.TITLE  PIO2.MAC

;  This program demonstrates the ability of the PIO to transfer
;  data on a byte-by-byte basis. The program uses the Interlocked
;  Handshake to transfer data from Port A to Port B. Both ports
;  are configured as single-buffered. The PIO loopback connector
;  (part #H3022 or 54-16227) or a functional equivalent is
;  required to successfully run this program.
;
;  After this program has been assembled and linked on the
;  development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the program into the KXJ11-CA.
;  The program will execute as shown in the following example.
;
;  SET 2
;  LOAD PIO2.SAV
;  EXECUTE
;  !ODT
;  !
;  !001214
;  !1262/065151
;  !001264/066153
;  !001266/067155
;  !001270/070157
;  !001272/000377
;  !001274/065151
;  !001276/066153
;  !001300/067155
;  !001302/070157
;  !001304/000000
;  !^C
;  EXIT
;
;  This verifies that the contents of the output buffer (location
;  1262 were successfully transferred to the input buffer
;  (location 1274).
;

        ; Register Assignments

        MIC     ==  177000
        MCC     ==  177002

        PAVEC   ==  177004
        PASTAT  ==  177020
        PADATA  ==  177032
        PAMODE  ==  177100
        PAHDSH  ==  177102
        PAPOL   ==  177104
        PASIO   ==  177110
```

```
     PBVEC   ==  177006
     PBSTAT  ==  177022
     PBDATA  ==  177034
     PBMODE  ==  177120
     PBHDSH  ==  177122
     PBPOL   ==  177124
     PBSIO   ==  177130

     PCPOL   ==  177012
     PCDDIR  ==  177014

     IOCNTL  ==  177140

START::
     MTPS    #340            ; Inhibit recognition of interrupts

     MOVB    #1,MIC          ; Reset device and inhibit interrupt
                             ; requests from the PIO
     CLRB    MIC             ; Enable device (interrupts still
                             ; inhibited)

     MOVB    #200,PAVEC
     MOV     #OUT,@#200      ; Set up Port A interrupt vector
     MOV     #340,@#202      ; ... and PSW

     MOVB    #204,PBVEC
     MOV     #IN,@#204       ; Set up Port B interrupt vector
     MOV     #340,@#206      ; ... and PSW

     ; Set-up Port A
     MOVB    #220,PAMODE     ; Port A: Output Port,
                             ; single-buffered
     CLRB    PAHDSH          ; Use interlock handshake
     CLRB    PAPOL           ; Port A bits are non-inverting
     CLRB    PASIO           ; Normal output
     MOVB    #300,PASTAT     ; Enable Port A interrupts

     ; Set-up Port B
     MOVB    #120,PBMODE     ; Port B: Input Port,
                             ; single-buffered
     CLRB    PBHDSH          ; Use interlock handshake
     CLRB    PBPOL           ; Port B bits are non-inverting
     CLRB    PBSIO           ; Normal input
     MOVB    #300,PBSTAT     ; Enable Port B interrupts

     ; Set-up the Port C handshake lines.
     ;  All handshake lines are configured as inputs, even
     ;  if they are not inputs!
     MOVB    #377,PCDDIR     ; Port C bits are inputs

     ; Set-up the PIO buffers
     MOV     #165400,IOCNTL  ; configure the PIO buffers for A=out
                             ; B=input, C0,C2=input, C1,C3=output
```

```
        ; Set-up data areas
        MOV     #OUTBUF,R0          ; Point to Output Buffer
        MOV     #INBUF,R1           ; Point to Input Buffer

        ; Enable Interrupts
        MOVB    #224,MCC            ; Enable ports A, B, and C
        MOVB    #200,MIC            ; Enable MIC
        MTPS    #0                  ; Enable recognition of interrupts

        ; Start the first transfer
        MOVB    #200,PASTAT         ; Set IP to initiate a transfer

        BR      .                   ; Wait here for the interrupts

    OUT::
        TSTB    (R0)                ; IF (R0) are negative
        BMI     1$                  ; THEN transfers are complete
                                    ; ELSE transfer another byte
        MOVB    (R0)+,PADATA        ; Move byte to the Port A output data
                                    ; register
        BR      2$
  1$:   MOVB    #240,PASTAT         ; Clear IP when done
  2$:   MOVB    #140,PASTAT         ; Clear IUS on each pass
        RTI

    IN::
        MOVB    PBDATA,(R1)+        ; Move byte from Port B input data
                                    ; register
        MOVB    #140,PBSTAT         ; Clear IUS on each pass
        RTI

OUTBUF: .BYTE   151,152,153,154,155,156,157,160,-1
        .EVEN
INBUF:  .BLKB   10

        .END    START
```

```
.TITLE  PIO3.MAC

;  This program is basically the same as PIO2.MAC with the
;  exception of double-buffered ports in this program. The
;  PIO loopback connector (part #H3022 or 54-16227) or a
;  functional equivalent is required to successfully run this
;  program. After this program has been assembled and linked on
;  the development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the program into the KXJ11-CA.
;  The program will execute as shown in the following example.
;
;  SET 2
;  LOAD PIO3.SAV
;  EXECUTE
;  !ODT
;  !
;  !001214
;  !1272/065151
;  !001274/066153
;  !001276/067155
;  !001300/070157
;  !001302/000377
;  !001304/065151
;  !001306/066153
;  !001310/067155
;  !001312/070157
;  !001314/000000
;  !^C
;  EXIT
;
;  This verifies that the contents of the output buffer (location
;  1272) were successfully transferred to the input buffer
;  (location 1304).

    ; Register Assignments

    MIC     ==  177000
    MCC     ==  177002

    PAVEC   ==  177004

    PASTAT  ==  177020
    PADATA  ==  177032
    PAMODE  ==  177100
    PAHDSH  ==  177102
    PAPOL   ==  177104
    PASIO   ==  177110

    PBVEC   ==  177006
    PBSTAT  ==  177022
```

```
   PBDATA  ==  177034
   PBMODE  ==  177120
   PBHDSH  ==  177122
   PBPOL   ==  177124
   PBSIO   ==  177130

   PCPOL   ==  177012
   PCDDIR  ==  177014

   IOCNTL  ==  177140

START::
   MTPS    #340                ; Inhibit recognition of interrupts
   MOVB    #1,MIC              ; Reset device and inhibit interrupt
                               ; requests from the PIO
   CLRB    MIC                 ; Enable device (interrupts still
                               ; inhibited)


   MOVB    #200,PAVEC
   MOV     #OUT,@#200          ; Set up Port A interrupt vector
   MOV     #340,@#202          ; ... and PSW

   MOVB    #204,PBVEC
   MOV     #IN,@#204           ; Set up Port B interrupt vector
   MOV     #340,@#206          ; ... and PSW

   ; Set-up Port A
   MOVB    #240,PAMODE         ; Port A: Output Port,
                               ; double-buffered
   CLRB    PAHDSH              ; Use interlock handshake
   CLRB    PAPOL               ; Port A bits are non-inverting
   CLRB    PASIO               ; Normal output
   MOVB    #300,PASTAT         ; Enable Port A interrupts

   ; Set-up Port B
   MOVB    #140,PBMODE         ; Port B: Input Port, double-buffered
   CLRB    PBHDSH              ; Use interlock handshake
   CLRB    PBPOL               ; Port B bits are non-inverting
   CLRB    PBSIO               ; Normal input
   MOVB    #300,PBSTAT         ; Enable Port B interrupts

   ; Set-up the Port C handshake lines.
   ;  All handshake lines are configured as inputs - even
   ;  if they are not inputs!
   MOVB    #377,PCDDIR         ; Port C bits are inputs

   ; Set-up the PIO buffers
   MOV     #165400,IOCNTL      ; configure the PIO buffers for A=out
                               ; B=input, C0,C2=input, C1,C3=output

   ; Set-up data areas
   MOV     #OUTBUF,R0          ; Point to Output Buffer
   MOV     #INBUF,R1           ; Point to Input Buffer
```

```
      ; Enable Interrupts
      MOVB     #224,MCC          ; Enable ports A, B, and C
      MOVB     #200,MIC          ; Enable MIC
      MTPS     #0                ; Enable recognition of interrupts

      ; Start the first transfer
      MOVB     #200,PASTAT       ; Set IP to initiate a transfer

      BR       .                 ; Wait here for the interrupts

   OUT::
      TSTB     (R0)              ; IF (R0) are negative
      BMI      1$                ; THEN transfers are complete
                                 ; ELSE transfer another byte
      MOVB     (R0)+,PADATA      ; Move 1st byte to the Port A output
                                 ; data register
      MOVB     (R0)+,PADATA      ; Move 2nd byte to the Port A buffer
                                 ; register
      BR       2$
1$:   MOVB     #240,PASTAT       ; Clear IP when done
2$:   MOVB     #140,PASTAT       ; Clear IUS on each pass
      RTI

   IN::
      MOVB     PBDATA,(R1)+      ; Move 1st byte from Port B input
                                 ; data register
      MOVB     PBDATA,(R1)+      ; Move 2nd byte from Port B buffer
                                 ; register
      MOVB     #140,PBSTAT       ; Clear IUS on each pass
      RTI

OUTBUF: .BYTE   151,152,153,154,155,156,157,160,-1
        .EVEN
INBUF:  .BLKB   10

        .END    START
```

```
.TITLE  PIO4I.MAC

;  The following is a more practical example - one KXJ11-CA
;  transferring data to another. Two programs follow: one
;  accepts data through Port B using the double-buffered mode
;  (PIO4I.MAC); the second program sends data out of Port A
;  using the double buffered mode (PIO4O.MAC).  In order to
;  successfully run these programs, the KXJ11-CA must be connected
;  by a "straight-thru" ribbon cable that is given a half twist.
;  In other words, it should make the same connections that the
;  PIO loopback connector does.
;
;  (A1-B1,A2-B2,...A7-B7,C0-C3,C1-C2).
;
;  Each program should be assembled and linked separately on the
;  development machine. Then use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the programs into the KXJ11-CA.
;  The program will execute as shown in the following example.
;
;  SET 3
;  LOAD PIO4I.SAV
;  EXECUTE
;  SET 2
;  LOAD PIO4O.SAV
;  EXECUTE
;  SET 3
;  !ODT
;  !
;  !001130
;  !1152/065151
;  !001154/066153
;  !001156/067155
;  !001160/070157
;  !001162/000000
;  !^C
;  EXIT
;
;  This verifies that the data was successfully transferred to
;  the input buffer of KXJ11-CA #3.
;
;-----------------------------------------------------------------
;-----------------------------------------------------------------

     ; Register Assignments

     MIC      ==  177000
     MCC      ==  177002

     PBVEC    ==  177006
     PBSTAT   ==  177022
     PBDATA   ==  177034
     PBMODE   ==  177120
     PBHDSH   ==  177122
```

```
        PBPOL    ==   177124
        PBDDIR   ==   177126
        PBSIO    ==   177130

        PCDDIR   ==   177014

        IOCNTL   ==   177140

START::
        MTPS     #340              ; Inhibit recognition of interrupts
        MOVB     #1,MIC            ; Reset device and inhibit interrupt
                                   ; requests from the PIO
        CLRB     MIC               ; Enable device (interrupts still
                                   ; inhibited)

        MOVB     #204,PBVEC
        MOV      #IN,@#204         ; Set up Port B interrupt vector
        MOV      #340,@#206        ; ... and PSW

        MOVB     #140,PBMODE       ; Port B: Input Port, double-buffered
        CLRB     PBHDSH            ; Use interlock handshake
        CLR      PBPOL             ; Port B bits are non-inverting
        CLR      PBSIO             ; Normal input
        MOVB     #300,PBSTAT       ; Enable Port B interrupts

        MOVB     #377,PCDDIR       ; Port C bits are inputs

        MOV      #165400,IOCNTL    ; configure the PIO buffers for A=out
                                   ; B=input, C0,C2=input, C1,C3=output

        MOV      #INBUF,R1         ; Point to input data buffer

        MOVB     #220,MCC          ; Enable ports B and C
        MOVB     #200,MIC          ; Enable MIC

        MTPS     #0                ; Enable recognition of interrupts
        BR       .                 ; Wait here for the interrupts

IN::
        MOVB     PBDATA,(R1)+      ; Move 1st byte from Port B input
                                   ; data register
        MOVB     PBDATA,(R1)+      ; Move 2nd byte from Port B buffer
                                   ; register
        MOVB     #140,PBSTAT       ; Clear IUS on each pass
        RTI

INBUF:   .BLKB    10

         .END     START
```

```
.TITLE  PIO40.MAC

        ; Register Assignments

        MIC      ==   177000
        MCC      ==   177002

        PAVEC    ==   177004
        PASTAT   ==   177020
        PADATA   ==   177032
        PAMODE   ==   177100
        PAHDSH   ==   177102
        PAPOL    ==   177104
        PADDIR   ==   177106
        PASIO    ==   177110

        PCPOL    ==   177012
        PCDDIR   ==   177014

        IOCNTL   ==   177140

START::
        MTPS     #340                 ; Inhibit recognition of interrupts
        MOVB     #1,MIC               ; Reset device and inhibit interrupt
                                      ; requests from the PIO
        CLRB     MIC                  ; Enable device (interrupts still
                                      ; inhibited)

        MOVB     #200,PAVEC
        MOV      #OUT,@#200           ; Set up Port A interrupt vector
        MOV      #340,@#202           ; ... and PSW

        MOVB     #240,PAMODE          ; Port A: Output Port,
                                      ; double-buffered
        CLRB     PAHDSH               ; Use interlock handshake
        CLR      PAPOL                ; Port A bits are non-inverting
        CLR      PASIO                ; Normal output
        MOVB     #300,PASTAT          ; Enable Port A interrupts

        MOVB     #377,PCDDIR          ; Port C bits are inputs

        MOV      #165400,IOCNTL       ; configure the PIO buffers for A=out
                                      ; B=input, C0,C2=input, C1,C3=output

        MOV      #OUTBUF,R0           ; Point to output data buffer

        MOVB     #24,MCC              ; Enable ports A  and C
        MOVB     #200,MIC             ; Enable MIC
        MTPS     #0                   ; Enable recognition of interrupts
        MOVB     #200,PASTAT          ; Set IP to initiate a transfer
        BR       .                    ; Wait here for the interrupts
```

```
OUT::
     TSTB    (R0)                ; IF (R0) are negative
     BMI     1$                  ; THEN all data has been transferred
                                 ; ELSE do another transfer
     MOVB    (R0)+,PADATA        ; Move 1st byte to the Port A output
                                 ; data register
     MOVB    (R0)+,PADATA        ; Move 2nd byte to the Port A buffer
                                 ; register
     BR      2$
1$:  MOVB    #240,PASTAT         ; Clear IP when done
2$:  MOVB    #140,PASTAT         ; Clear IUS on each pass
     RTI

OUTBUF::
     .BYTE   151,152,153,154,155,156,157,160,-1

     .END    START
```

```
;  The following two programs demonstrate how the DTC may be used
;  to transfer data from the PIO to KXJ11-CA local memory.  DTC
;  transfers may only be accomplished using Port A of the PIO.
;  It is not possible to properly connect two PIOs with a ribbon
;  cable, because the handshake lines will not align correctly
;  when connecting Port A to Port A. Therefore, it is necessary to
;  build a cable that makes the following connections.
;
;           Input Port A                Output Port A
;                A0      <--------->         A0
;                A1      <--------->         A1
;                 .                           .
;                 .                           .
;                A7      <--------->         A7
;
;                C2      <--------->         C3
;                C3      <--------->         C2
;
;  It is also necessary to place a jumper between posts M48 and
;  M49 so that the REQUEST line from the PIO may signal the DTC.
;  For more information about programming the DTC, please refer to
;  Section 4.3.
;
;  After each program has been assembled and linked on the
;  development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the programs into a KXJ11-CA.
;  The program will execute as shown in the following example.
;
;  SET 3
;  LOAD PIO5I.SAV
;  EXECUTE
;  SET 2
;  LOAD PIO5O.SAV
;  EXECUTE
;  SET 3
;  !ODT
;  !
;  !001140
;  !1140/000777
;  !001142/065151
;  !001144/066153
;  !001146/067155
;  !001150/070157
;  !001152/001602
;  !^C
;
;  Examining the contents of the input buffer (location 1142)
;  verifies that the data was successfully transferred.
```

```
.TITLE  PIO5I.MAC

; This program transfers data from Port A of the PIO to local
; memory by utilizing the DTC

     ; Register Assignments

     MMREG   ==  174470
     CMDREG  ==  174454
     CASTFl  ==  174444
     CAOFl   ==  174440

     MIC     ==  177000
     MCC     ==  177002

     PAVEC   ==  177004
     PASTAT  ==  177020
     PADATA  ==  177032
     PAMODE  ==  177100
     PAHDSH  ==  177102
     PAPOL   ==  177104
     PADDIR  ==  177106
     PASIO   ==  177110

     PCPOL   ==  177012
     PCDDIR  ==  177014

     IOCNTL  ==  177140

START::
     MTPS    #340            ; Inhibit recognition of interrupts

     ; Initialize the DTC - for more information on the DTC
     ; refer to Section 4.3.
     MOVB    #154,MMREG      ; Load Master Mode Reg to Disable DTC
     CLRB    CMDREG          ; Reset the DTC
     MOV     #0,CASTFl       ; Load the CHl Register SEG/TAG
     MOV     #RELOAD,CAOFl   ; Load the CHl Register Offset
     MOVB    #155,MMREG      ; Load Master Mode Reg to Enable DTC
     MOVB    #241,CMDREG     ; Start Chain Channel 1

     ; Initialize the PIO
     MOVB    #1,MIC          ; Reset device and inhibit interrupt
                             ; requests from the PIO
     CLRB    MIC             ; Enable device (interrupts still
                             ; inhibited)
     ; Set-up Port A
     MOVB     #120,PAMODE       ; Port A: Input Port, single-buffered
     MOVB     #70,PAHDSH        ; Use interlock handshake, input
                                ; REQUEST
     CLR      PAPOL             ; Port A bits are non-inverting
     CLR      PASIO             ; Normal input
     MOVB     #2,PCPOL          ; Invert pin C1 - this is the line
                                ; that is used for the REQUEST
                                ; signal
     MOVB     #377,PCDDIR       ; Port C bits are inputs

     MOV      #164377,IOCNTL    ; configure the PIO buffers for A=in
                                ; B=output, C0,C2=input, C1,C3=output

     MOV      #INBUF,R1         ; Point to input data buffer

     MOVB     #24,MCC           ; Enable ports A and C

     BR       .                 ; Wait here while the DMA transfers
                                ; take place

    INBUF:  .BLKB   10

; Chain Load Region

RELOAD: .WORD   001602   ; Reload Word <Select CARA,CARB,COPC,CM>

      .WORD    000020           ; Current Address Register A Seg/Tag
      .WORD    padata+1         ; Current Address Register A Offset
                                ; <This local address is the source,
                                ; its address is held constant,
                                ; since the DTC is doing byte
                                ; transfers specify
                                ; the source address high byte>

      .WORD    000000           ; Current Address Register B Seg/Tag
      .WORD    inbuf            ; Current Address Register B Offset
                                ; <This local address is the
                                ; destination>

      .WORD    000010           ; Current Operation Count <Transfer 8
                                ; words>

      .WORD    000000           ; Channel Mode Register High
      .WORD    000001           ; Channel Mode Register Low
                                ; <No match conditions, do nothing
                                ; upon completion, transfer type =
                                ; Single Transfer CARA = source, byte
                                ; transfers>

      .END     START
```

```
.TITLE  PIO5O.MAC

;  This program transfers data out of Port A of the PIO
;  utilizing the DTC

; Register Assignments

    MMREG   ==  174470
    CMDREG  ==  174454
    CASTF1  ==  174444
    CAOF1   ==  174440

    MIC     ==  177000
    MCC     ==  177002

    PAVEC   ==  177004
    PASTAT  ==  177020
    PADATA  ==  177032
    PAMODE  ==  177100
    PAHDSH  ==  177102
    PAPOL   ==  177104
    PADDIR  ==  177106
    PASIO   ==  177110

    PCPOL   ==  177012
    PCDDIR  ==  177014

    IOCNTL  ==  177140

  START::
    MTPS    #340                ; Inhibit recognition of interrupts

    ; Initialize the DTC
    MOVB    #154,MMREG          ; Load Master Mode Reg to Disable DTC
    CLRB    CMDREG              ; Reset the DTC
    MOV     #0,CASTF1           ; Load the CH1 Register SEG/TAG
    MOV     #RELOAD,CAOF1       ; Load the CH1 Register Offset
    MOVB    #155,MMREG          ; Load Master Mode Reg to Enable DTC
    MOVB    #241,CMDREG         ; Start Chain Channel 1

    ; Initialize the PIO
    MOVB    #1,MIC              ; Reset device and inhibit interrupt
                                ; requests from PIO
    CLRB    MIC                 ; Enable device (interrupts still
                                ; inhibited)

    ; Set-up Port A
    MOVB    #220,PAMODE         ; Port A: Output Port,
                                ; single-buffered
    MOVB    #050,PAHDSH         ; Use interlock handshake, output
                                ; REQUEST
    CLR     PAPOL               ; Port A bits are non-inverting
    CLR     PASIO               ; Normal output
    MOVB    #2,PCPOL          ; Pin C1 must be inverted - this is
                              ; the line used to signal the DTC
    MOVB    #377,PCDDIR       ; Port C bits are inputs

    MOV     #165400,IOCNTL    ; configure the PIO buffers for A=out
                              ; B=input, C0,C2=input, C1,C3=output

    MOV     #OUTBUF,R0        ; Point to output data buffer

    MOVB    #24,MCC           ; Enable ports A and C

    BR                        ; Wait here while the DMA transfers
                              ; complete

  OUTBUF::
    .BYTE   151,152,153,154,155,156,157,160,-1
    .EVEN

; CHAIN LOAD REGION

RELOAD: .WORD  001602   ; Reload Word <Select CARA,CARB,COPC,CM>

        .WORD  000000   ; Current Address Register A Seg/Tag
        .WORD  outbuf   ; Current Address Register A Offset
                        ; <This local address is the source>

        .WORD  000020   ; Current Address Register B Seg/Tag
        .WORD  padata+1; Current Address Register B Offset
                        ; <This local address is the destination,
                        ; Hold the address, must specify high byte
                        ; for byte transfer>

        .WORD  000010   ; Current Operation Count <Transfer 8
                        ; words>

        .WORD  000000   ; Channel Mode Register High
        .WORD  000001   ; Channel Mode Register Low
                        ; <No match conditions, do nothing upon
                        ; completion, transfer type = Single
                        ; Transfer
                        ; CARA = source, byte transfers>

        .END   START
```

5.3.3 Programming The PIO Counter/Timers

This section describes how to program the PIO Counter/Timers and provides example programs demonstrating their capabilities.

Each of the three PIO Counter/Timers provides up to four lines for external access. If these external lines are used, the corresponding port pins must be available and programmed in the proper direction. Table 5-5 displays which port pins correspond to the Counter/Timer external access lines:

The first step in programming a PIO Counter/Timer is to specify which, if any, external lines are to be used, the output duty cycle, and whether the cycle is continuous or single-cycle. Figure 5-22 displays the available output duty cycles.

Table 5-5 PIO Counter/Timer External Access Lines

Function	C/T 1	C/T 2	C/T 3
Counter/Timer Output	Port B4	Port B0	Port C0
Counter Input	Port B5	Port B1	Port C1
Trigger Input	Port B6	Port B2	Port C2
Gate Input	Port B7	Port B3	Port C3

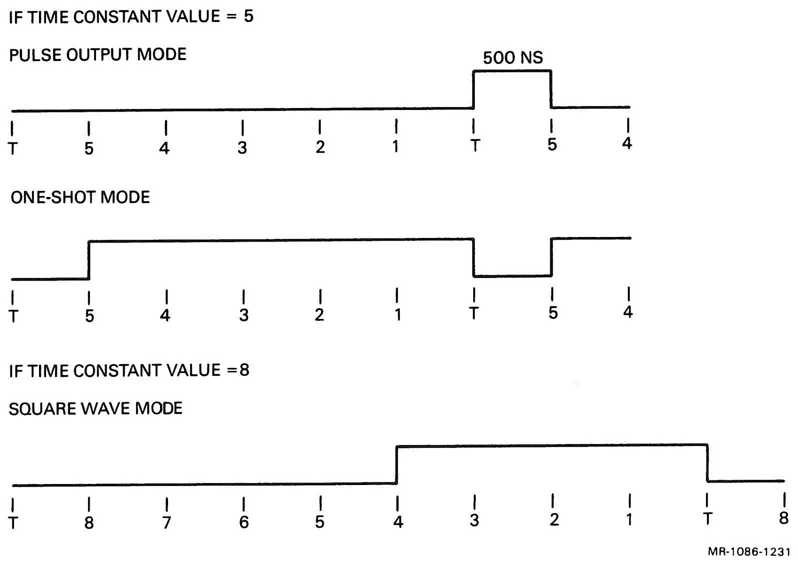


Figure 5-22 PIO Output Duty Cycles

Output Duty Cycles

Next, the Time Constant Registers must be loaded. Each Counter/Timer contains two of these registers, which are used to form the 16-bit value that is loaded into the down-counter when the Counter/Timer is triggered.

If external lines are to be used, then the corresponding port pins should be programmed as bit ports with the correct data direction. Finally, the Counter/Timer enable bit for that port must be enabled in the Master Configuration Control Register.

The down-counter is loaded and the countdown sequence is initiated when the Counter/Timer is triggered. This trigger may occur if the Trigger Command Bit (TCB) in the Command and Status Register is set, or if an external trigger input was asserted. Once the countdown is initiated, it will continue towards the terminal count as long as the Gate Command Bit (GCB) in the Command and Status Register is set and the Gate Input is held asserted (if it is enabled). If a trigger occurs during a countdown sequence, the action taken is determined by the Retrigger Enable Bit (REB). If REB = 0, the trigger is ignored, but if REB = 1, the down-counter is reloaded and a new countdown is initiated.

When the terminal count is reached, the state of the Continuous/Single Cycle bit (C/SC) in the Mode Specification Register is examined. If C/SC = 0, the countdown sequence stops. If C/SC = 1, the time constant is reloaded and a new countdown is initiated. If the Interrupt Enable Bit (IE) is set, an interrupt request is generated when the down-counter reaches its terminal count. If a terminal count occurs while the Interrupt Pending Bit (IP) is set, an error is indicated by the Interrupt Error (ERR) bit.

The following program provides an example of how to program the PIO Counter/Timers.

```
.TITLE  CT1.MAC

;  This program demonstrates how to utilize one of the
;  Counter/Timers on the KXJ11-CA. Counter/Timer 1 will be used in
;  this program. This counter/timer is clocked at a 500 ns rate.
;  The time constant used for the counter is 50,000. Therefore,
;  the countdown sequence will take 25 ms. (500 ns X 50,000 =
;  25,000,000 ns = 25 ms). The interrupt service routine waits
;  until the countdown sequence has completed 40 times and then
;  outputs an 'A' out of the console port. This should happen
;  approximately one time a second. (25 ms X 40 = 1 s).
;
;  After this program has been assembled and linked on the
;  development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the program into the KXJ11-CA.
;  The program will execute as shown in the following example.
```

```
;
;  SET 2
;  LOAD CTl.SAV
;  EXECUTE
;  EXIT
;
;  Notice that the 'A's keep on coming after you exit KUI!

     ; Register Assignments

     MIC     ==  177000
     MCC     ==  177002
     CTVEC   ==  177010
     CTlCON  ==  177024
     CTlHI   ==  177054
     CTlLO   ==  177056
     CTlMOD  ==  177070

START::
     MTPS    #340            ; Disable recognition of interrupts

     MOVB    #1,MIC          ; Reset PIO
     CLRB    MIC             ; Enable PIO (Interrupts disabled)

     MOVB    #210,CTVEC      ;
     MOV     #ISR,@#210      ; Initialize Counter/Timer vector
     MOV     #340,@#212      ; and ISR address

     CLR     Rl              ; Used as a counter

     MOVB    #200,CTlMOD     ; Select continuous mode, no external
                             ; access, pulse output
     MOVB    #203,CTlHI      ; CTlHI and CTlLO combine to form
     MOVB    #120,CTlLO      ; 141520(8) = 50000(10)

     MOVB    #100,MCC        ; Enable Counter/Timer l
     MOVB    #200,MIC        ; Enable PIO interrupts
     MTPS    #0              ; Enable recognition of interrupts

     BISB    #306,CTlCON     ; Set IE,GCB,TCB - this starts the
                             ; countdown
     BR      .               ; Wait here for the interrupts

ISR:
     INC     Rl              ; Increment the counter
     CMP     Rl,#40.         ; IF this is not the 40th time
     BNE     2$              ; THEN count again
     CLR     Rl              ; ELSE clear the counter and...
     MOVB    #101,@#177566   ; send an 'A' to the console
;+
;  The console in this case is the KXJll-CA console - NOT the
;  development system console.  Therefore, you'll have to hook a
;  terminal up  to SLUl to see the 'A's pop out.
;-
2$:  MOVB    #44,CTlCON      ; Clear IUS and IP but don't bother
                             ; GCB
     RTI

     .END    START
```

CHAPTER 6
SERIAL LINE UNITS

6.1 OVERVIEW

The KXJ11-CA has two serial line units (SLUs): SLU1 and SLU2.

SLU1 is also called the console port and is designed around the DLART (DC319) chip. SLU1 is dedicated for a console device. For details on the operation and programming of the DLART, refer to the DLART Data Sheet included as part of this documentation package.

SLU2 is also called the multiprotocol serial controller (MPSC) and has two independent channels, A and B. SLU2 is designed around the uPD7201 chip. For details on the operation of the MPSC, refer to the MPSC Data Sheet included in this documentation package. There are three timers associated with SLU2, and designed around the 8254 chip. Information on these timers is included here for completeness. Refer to the 8254 Data Sheet included as part of this documentation package for operational details.

The material that follows summarizes and describes the DLART and MPSC functions implemented on the KXJ11-CA.

6.2 CONSOLE SERIAL PORT (SLU1)

SLU1 provides the following features and capabilities for the console serial line:

Asynchronous operation
Error detection overrun, framing, and BREAK detection
Internal baud rate generation from 300 to 38.4 K baud
Common baud rate for both transmitter and receiver
50- and 60-Hz real-time clock interrupt outputs
One stop bit only

6.2.1 SLU1 (Console) Registers

The console serial port (SLU1) is based on the DLART (DC319) chip. All SLU1 registers are contained in this chip. SLU1 has a receiver and a transmitter, each of which has a control/status register and a buffer register. These registers are described in the sections that follow. Note that these registers are the ones used for console ODT operations.

6.2.1.1 Receiver Control/Status Register (RCSR) -- The Receiver Control/Status Register (Figure 6-1) is used to monitor and control the operation of the SLU1 receiver. This register is read-only.

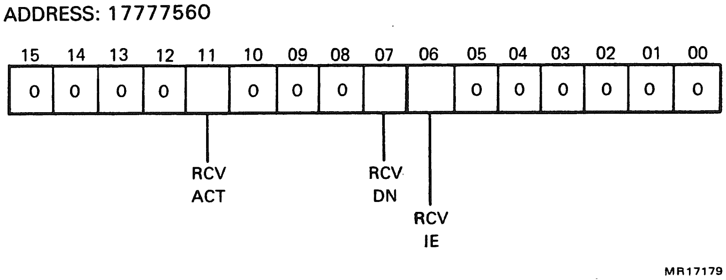


Figure 6-1 Receiver Control/Status Register (RCSR)

Bits	Name	Description
15:12		Not used (read as zeros)
11	RCV ACT	Receiver active - When set, the receiver is active. Set when the start bit of the input serial data is received. When cleared, the receiver is inactive. Cleared at the expected time of reception of the stop bit (after RCV DN is set).
10:8		Not used (read as zeros)
7	RCV DN	Set after a character has been received and is in the receiver buffer register (RBUF). Cleared when the character is read from RBUF.
6	RCV IE	When set, allows an interrupt request to be made when bit 7 (RCV DN) is set. When cleared, disables interrupts from RCV DN.
5:0		Not used (read as zeros)

6.2.1.2 Receiver Buffer Register (RBUF) -- The Receiver Buffer Register (RBUF) (Figure 6-2) holds the most recent byte received and contains break and error information for this byte. RBUF is read-only.

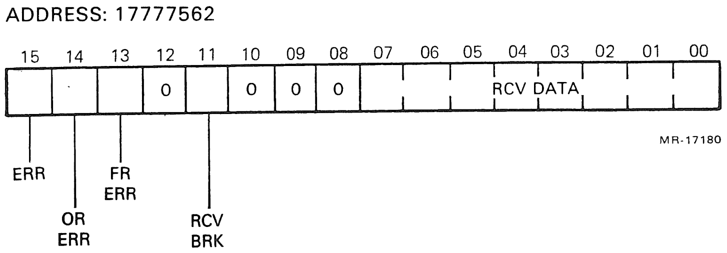


Figure 6-2 Receiver Buffer Register (RBUF)

Bits	Name	Description
15	ERR	Error - Set when bit 14 (OR ERR) or bit 13 (FR ERR) is set. Cleared when the condition causing the error is cleared.
14	OR ERR	Overrun error - Set when a received byte is loaded into bits 7:0 (RCV DATA) before bit 7 of the RCSR (RCV DN) is cleared. This occurs when a new byte is received before the reception of the previous byte is complete. This bit is updated each time a byte is received.
13	FR ERR	Framing error - Set when a received byte is loaded into bits 7:0 (RCV DATA) without a valid stop bit. FR ERR is updated each time a byte is received.
12		Not used (read as zero)
11	RCV BRK	Receive break - Set when the receiver's serial input line goes from a mark to a space condition and stays in the space condition for 11 bit times after reception starts. Cleared when serial input returns to the mark condition.
10:8		Not used (read as zeros)
7:0	RCV DATA	Receive data - Contains the most recent byte received. Each time a byte is received, the RCV DN bit in the RCSR is set.

6.2.1.3 **Transmitter Control/Status Register (XCSR)** -- The Transmitter Control/Status Register (Figure 6-3) is used to monitor and control the operation of the SLU1 transmitter. Bits <15:7> of this register are read-only. Bits <6:0> are read/write.

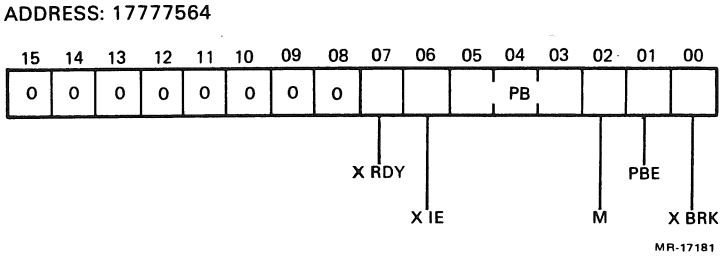


Figure 6-3 Transmitter Control/Status Register (XCSR)

Bits	**Name**	**Description**
15:8		Not used (read as zeros)
7	X RDY	Transmit ready - When set, the Transmitter Buffer Register (XBUF) is ready to accept a byte. Cleared when XBUF is written.
6	X IE	Transmit interrupt enable - When set, allows an interrupt request to be made when bit 7 (X RDY) is set. When cleared, disables interrupts from X RDY.
5:3	PB	Programmable baud rate - These bits determine the transmitter and receiver baud rate as shown:

PB	**Baud Rate**
000	300
001	600
010	1200
011	2400
100	4800
101	9600
110	19200
111	38400

2	M	Maintenance - When set, external serial data input to SLU1 is disabled and the transmitter serial output is connected to the receiver serial input. This allows self-testing of SLU1.

Bits	Name	Description
1	PBE	Programmable baud rate - When set, the baud rate is determined by PB. When cleared, the baud rate is determined by a source external to SLU1.
0	X BRK	Transmit break - When set, the serial output line is forced to a space condition.

6.2.1.4 Transmitter Buffer Register (XBUF) -- The Transmitter Buffer Register (Figure 6-4) holds the most recent byte transmitted. Bits <15:8> of this register are read-only. Bits <7:0> are read/write.

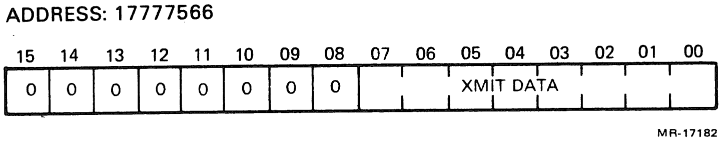


Figure 6-4 Transmitter Buffer Register (XBUF)

Bits	Name	Description
15:8		Not used (read as zeros)
7:0	XMIT DATA	Transmit data - Contains the next byte to be transmitted. When the X RDY bit in the XCSR is clear, XMIT DATA is copied into a shift register (the serial data output register) for transmission. XMIT DATA is loaded with the next byte to be transmitted, which sets X RDY. When X RDY is cleared, the operation is repeated.

6.3 MULTIPROTOCOL SERIAL CONTROLLER (SLU2)

SLU2 provides the KXJ11-CA with the following features and capabilities.

- Two full duplex channels

  Channel A provides full modem control
  Channel B provides data and timing leads only

- Each channel may be operated in one of three modes

  Asynchronous
  5, 6, 7, or 8 Data bits
  1, 1-1/2, or 2 Stop bits
  Odd, Even, or No Parity
  Break generation and detection
  Interrupt on Parity, Overun, or Framing Errors

  Character-oriented synchronous
  Monosync, Bisync, and External Sync Operations
  Software Selectable Sync Characters
  Automatic Sync Insertion
  CRC Generation and Checking

  Bit-oriented synchronous
  HDLC and SDLC Operations
  Abort Sequence Generation and Detection
  Automatic Zero Insertion and Detection
  Address Field Recognition
  CRC Generation and Checking
  I-Field Residue Handling

- Programmable Baud Rates

- Double Buffered Transmitted Data

- Quadruple-Buffered Received Data

- Programmable CRC Algorithm

- Channel A may utilize the DMA controller to transfer data

6.3.1 Synchronous/Asynchronous Serial Line (SLU2) Registers

SLU2 is a synchronous/asynchronous serial device with two independent channels, A and B. SLU2 is based on the uPD7201 chip. The registers associated with SLU2 are summarized in Table 6-1.

KXJ11 Control/Status Register A is contained in the GAS on-board gate array. The timer data and control registers are contained in an on-board Intel 8254-2 timing controller chip. The other registers are contained in SLU2 itself, that is, the uPD7201 chip.

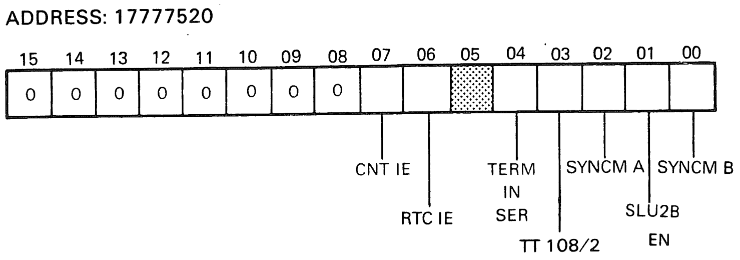
Table 6-1 SLU2 Registers

Address	Access	Description
17777520	RW	KXJ11 Control/Status Register A
17775736	W	SLU2 Timer Control Register
17775734	W	SLU2 Timer 2 Data Register
17775732	W	SLU2 Timer 1 Data Register
17775730	W	SLU2 Timer 0 Data Register
17775724	R	SLU2 Timer 2 Data Register
17775722	R	SLU2 Timer 1 Data Register
17775720	R	SLU2 Timer 0 Data Register

Channel A Address	Channel B Address	Access	Description
17775706	17775716	W	Transmitter
17775704	17775714	W	Control Register
17775702	17775712	R	Receiver
17775700	17775710	R	Status Register

6.3.1.1 KXJ11 Control/Status Register A (KXJCSRA) -- This register contains control information which affects the overall operation of SLU2. The register is cleared whenever the KXJ11-CA is powered up or reinitialized.

KXJ11 Control/Status Register A has the following format (Figure 6-5).



MR-17146

Figure 6-5 KXJ11 Control/Status Register A

Bits	Name	Description
15:8		Not used (read as ones)
7	CNT IE	Programmable counter interrupt enable - When set, interrupts from programmable timer/counter 2 are enabled. When cleared, these interrupts are inhibited.
6	RTC IE	Real-time clock interrupt enable - When set, interrupts from the on-board real-time clock (RTC) are enabled. When cleared, these interrupts are disabled.
5		Not used (read/write)
4	TERM IN SER	Terminal in service - For use with modems. When set, Terminal In Service (IS) is asserted and incoming calls can be connected. When cleared, IS is not asserted.
3	TT108/2	Modem connected - For use with modems. When set, Terminal Ready (TR) is asserted. When cleared, TR is not asserted.
2	SYNCM A	Clock select channel A - When set, SLU2 channel A receives its clock from the on-board baud rate generator. When cleared, channel A receives its clock from an external source.
1	SLU2BR EN	Party line enable - Used when the KXJ11-CA is configured for party line operation. When set, SLU2 channel B can not receive party line data. When cleared, party line data reception for channel B is enabled.
0	SYNCM B	Clock select channel B - When set, SLU2 channel B receives its clock from the on-board baud rate generator. When cleared, channel B receives its clock from an external source.

6.3.1.2 Timer Registers -- There are three independent timers associated with SLU2. These timers, labeled 0, 1, and 2, are contained in an on-board 8254-2 timing controller chip. Timer 0 and timer 1 run at 9.8304 MHz and are used to determine the baud rates for SLU2 channels A and B, respectively. Timer 2 is a general-purpose 800 Hz clock capable of generating interrupts at priority level 6. Interrupts from the 800 Hz timer are enabled and disabled via bit 7 of KXJ11-CA Control Register A (see Section 6.3.1.1).

Each timer has a Control Register and a Data Register. To use a timer, its Control Register is loaded first with configuration information. Then, its Data Register is loaded with the number of clock "ticks" the timer is to count.

The baud rates for channels 0 and 1 can be set by loading a "divider ratio" into a Data Register. For synchronous transmission,

Divider ratio = 9830.4 K / synchronous baud rate

Table 6-2 illustrates the correspondence between various divider ratios and synchronous baud rates.

For asynchronous transmission,

Divider ratio = 614.4 K / asynchronous baud rate

Table 6-3 illustrates the correspondence between various divider ratios and asynchronous baud rates.

Table 6-2 Synchronous Baud Rates

Divider Ratio (Decimal)	Synchronous Baud Rate
136	72.282 K
176	55.855 K
204	47.720 K
512	19.200 K
1024	9.600 K
2048	4.800 K
8192	1.200 K

Table 6-3 Asynchronous Baud Rates

Divider Ratio (Decimal)	Synchronous Baud Rate
2	307.2 K
4	153.6 K
8	76.8 K
16	38.4 K
32	19.2 K
64	9.6 K
128	4.8 K
256	2.4 K
512	600.0
1024	300.0
2048	150.0
5586	109.989

6.3.1.2.1 SLU2 Timer Control Registers -- There are three Timer Control Registers, one for each timer. They all have the format shown in Figure 6-6.

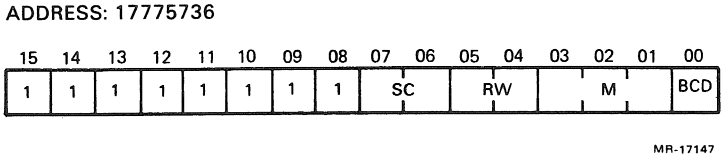


Figure 6-6 Timer Control Register Format (Timers 0, 1, and 2)

Bits **Name** **Description**

15:8 Not used (read as ones)

7:6 SC Select counter - Determines which counter is selected or whether a read back command is issued.

SC	Selection
00	Select counter 0
01	Select counter 1
10	Select counter 2
11	Issue read-back command

If a read-back command is issued, bits <5:0> of the Timer Control Register are defined as follows:

Bit **Definition**

5 Count - When set, latches the contents of the Timer Counter Data Register(s) specified by bits <3:1>. The contents of the register(s) are interpreted as a count of clock "ticks."

4 Status - When clear, latches the contents of the Timer Counter Data Register(s) specified by bits <3:1>. The contents of the register(s) are interpreted as status information.

3 When set, specifies counter 2

2 When set, specifies counter 1

1 When set, specifies counter 0

0 Must be zero

Bits	Name	Description
5:4	RW	Read/write - Determines which byte of information is read/written to/from a Timer Data Register or whether a counter latch command is issued. RW Selection 00 Issue counter latch command 01 Read/write least significant byte only 10 Read/write most significant byte only 11 Read/write least significant byte first, then most significant byte. If a counter latch command is issued, bits <3:0> of the Timer Control Register are "don't care" bits and are not interpreted.
3:1	M	Mode select - Selects the operational mode of the timer. See the uPD7201 Data Sheet for descriptions of these modes. M Mode 000 Reserved 001 Reserved 010 Baud rate generator 011 Square wave 100 Software triggered strobe 101 Reserved 110 Reserved 111 Reserved
0	BCD	BCD enable - When set, indicates that the information in the Timer Data Register is to be interpreted in binary coded decimal (BCD) format (four decades). When cleared, the data is interpreted in 16-bit binary format.

6.3.1.2.2 SLU2 Timer Data Registers -- There are six Timer Data Registers, two for each timer. Each timer has one register for read data and another register for write data. They all have the format shown in Figure 6-7 except when status data is read. In that case, the format is as shown in Figure 6-8.

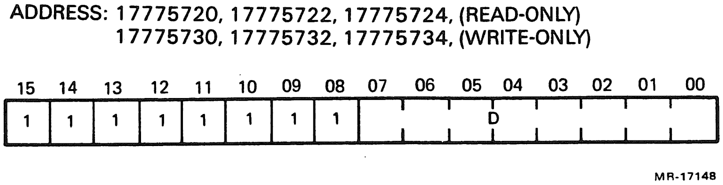

ADDRESS: 17775720, 17775722, 17775724, (READ-ONLY)
17775730, 17775732, 17775734, (WRITE-ONLY)
15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
1 1 1 1 1 1 1 1
D
MR-17148

Figure 6-7 Timer Data Register Format (Read and Write Registers 0, 1, and 2)

Bits	Name	Description
15:8		Not used (read as ones)
7:0	D	Counter data - specifies a number of timer clock "ticks."

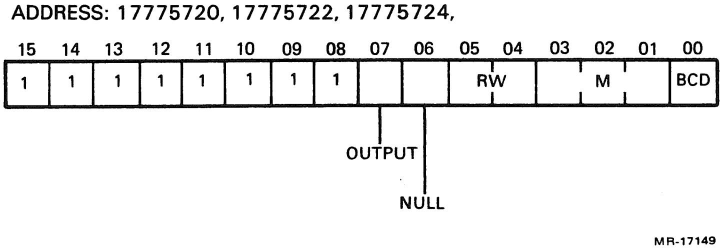


Figure 6-8 Timer Data Register Format When Used as a Timer Status Register

Bits	Name	Description
15:8		Not used (read as ones)
7	OUTPUT	Output - When set, the corresponding timer output signal is asserted. The state of this bit is the same as the state of the timer's pin on the 8254-2 chip.
6	NULL	Null - When cleared, a new count has been written and is ready to be read. When set, the counter contains a "null count" value which should not be read unless the user desires the previous (not updated) count.

Bits	Name	Description
5:4	RW	Read/write - Determines which byte of information is read/written to/from a Timer Data Register or whether a counter latch command is issued. **RW Selection** 00 Issue counter latch command 01 Read/write least significant byte only 10 Read/write most significant byte only 11 Read/write least significant byte first, then most significant byte If a counter latch command is issued, bits <3:0> of the Timer Control Register are "don't care" bits and are not interpreted.
3:1	M	Mode select - Selects the operational mode of the timer. **M Mode** 000 Reserved 001 Reserved 010 Baud rate generator 011 Square wave 100 Software triggered strobe 101 Reserved 110 Reserved 111 Reserved
0	BCD	BCD enable - When set, indicates that the information in the Timer Data Register is to be interpreted in binary coded decimal (BCD) format (four decades). When cleared, the data is interpreted in 16-bit binary format.

6.3.1.2.3 SLU2 Timer Programming Considerations -- When the watchdog timer is operating in Mode 2 (that is, as a baud rate generator), an initial count is loaded into the Timer Data Register and the count is decremented upon each clock tick. When the count is decremented to a value of 1, an interrupt is generated, the initial count is reloaded, and the count is again decremented. This process causes the watchdog timer to periodically generate an interrupt.

When a read-back command is issued to the watchdog timer, the timer is loaded on the next 800 Hz clock tick. If the user tries to read back values written into the timer before the clock tick has occurred, erroneous data may be read. The user should delay the reading back of values written to the timer for at least 1.25 milliseconds.

The way to stop the watchdog timer is to write the Timer Control Register, selecting counter 2 in bits <7:6> and specifying a valid mode in bits <3:1>. When this is done, the timer stops and waits for a new initial count to be loaded into the Timer Data Register. If a new initial count is not loaded, the operation of the timer is suspended and effectively stopped.

Hardware resets have no effect on the operation of the watchdog timer. A hardware reset only clears a pending interrupt (if any) from the timer.

6.3.1.3 SLU2 Control Registers -- Each channel has a set of eight write-only Control Registers numbered 0 through 7. Control Register 0 can be written directly. Control Registers 1 through 7 are accessed by first writing Control Register 0 bits <2:0> and then writing the desired Control Register. This section describes each of the Control Registers.

6.3.1.3.1 Control Register 0 -- See Figure 6-9.

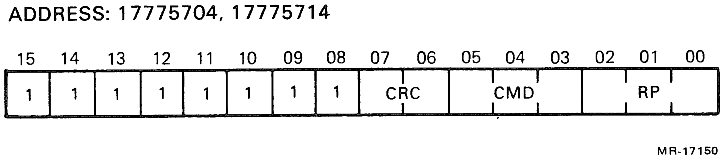


Figure 6-9 Control Register 0

Bits	Name	Description
15:8		Not used (read as ones)
7:6	CRC	CRC control - The following commands control the operation of the cyclic redundancy check (CRC) circuitry:
		CRC **Mode**
		00 Null - No effect. Used when setting other fields in Control Register 0 such as the register pointer field.
		01 Reset receiver CRC checker - In synchronous mode, resets the CRC checker to zeros. In SDLC mode, resets the CRC checker to ones.
		10 Reset transmitter CRC generator - In synchronous mode, resets the CRC generator to zeros. In SDLC mode, resets the CRC generator to ones.
		11 Reset idle/CRC latch - Clears the idle/CRC latch. When a transmitter underrun occurs, the transmitter enters the CRC phase of operation and begins to send the CRC character calculated up to that point. Then the latch is set. If the underrun condition persists, idle characters are sent after the CRC character. This latch is set when the channel is initialized.
5:3	CMD	Commands - The following SLU2 commands are specified by this field:
		CMD **Command**
		000 Null - No effect. Used when setting other fields in Control Register 0 such as the register pointer and the CRC command field.
		001 Send abort - Used in SDLC mode. Causes an SDLC abort code to be transmitted.
		010 Reset external/status interrupts - Clears any pending external interrupts and allows new interrupts to be detected.

Bits	Name	Description
		011 Channel reset - Disables the channel's receivers and transmitters (transmitter outputs are set high) and sets modem control outputs high. Disables interrupts and clears all DMA and interrupt requests. All Control Registers must be rewritten after a channel reset command. One NOP instruction must be executed before a new command can be written.
		100 Enable interrupt on next character - Used when operating in Interrupt on First Character mode. Reenables the interrupt logic for the next received character.
		101 Reset pending transmitter interrupt/DMA request - Clears a pending Transmitter Buffer Becoming Empty interrupt or DMA request without sending another character.
		110 Error reset - Clears a Special Receive Condition interrupt. Clears parity and overrun errors.
		111 End of interrupt (Channel A only) - Typically included as part of an interrupt service routine. Reenables lower priority devices in the interrupt daisy chain for servicing of any pending interrupts.
2:0	RP	Register pointer - Specifies which Control Register will be written or which Status Register will be read next. When the KXJ11-CA is reset or initialized, this field is set to 000 which allows the writing of Control Register 0 or the Reading of Status Register 0. Following a read or write to a Control Register other than 0, this field is set to 000.

6.3.1.3.2 Control Register 1 -- See Figure 6-10.

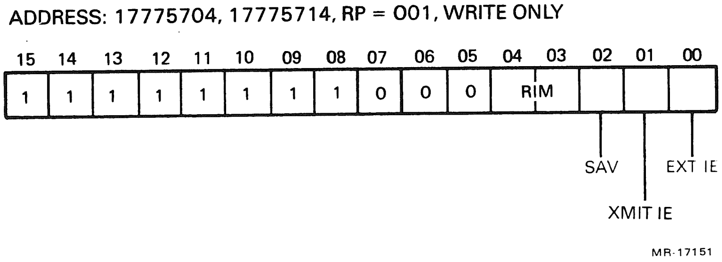


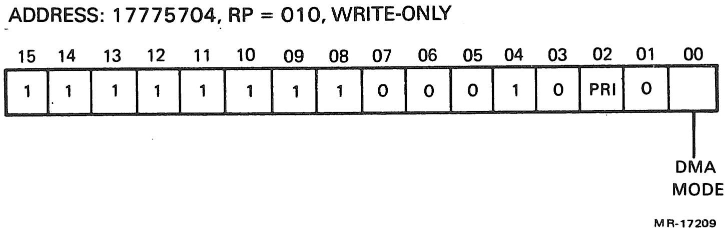
Figure 6-10 Control Register 1

Bits	Name	Description
15:8		Not used (read as ones)
7:5		Must be zero
4:3	RIM	Receiver interrupt mode - Determines how a channel handles received characters.

RIM	Interrupt Mode
00	Receiver interrupts/DMA request disabled - Disables interrupt or DMA requests from this channel if a character is received (polled mode).
01	Interrupt on first character only - Causes an interrupt to be issued for the first character received after an enable interrupt on first character command has been given (see description of Control Register 0). If the channel is in DMA mode, a DMA request is issued for each character received, including the first.
10	Interrupt on all received characters - Causes an interrupt to be issued whenever a character is present in the channel's receive buffer. A DMA request is issued if the channel is in DMA mode. A parity error is considered a Special Receive Condition.
11	Interrupt on all received characters - This is similar to 10, described previously. The difference is that a parity error is not considered to be a Special Receive Condition.

Bits	Name	Description
2	SAV	Status affects vector - Must be 1 for channel B, must be 0 for channel A. This setting ensures that the vector loaded into Status Register 2, channel B, is modified to indicate the cause of the interrupt.
1	XMIT IE	Transmit interrupt enable - When set, this channel will issue an interrupt when the transmitter buffer becomes empty or when the transmitter enters an Idle phase and begins transmitting sync or flag characters.
0	EXT IE	External status interrupt enable - When set, this channel will issue an interrupt when any of the following occur: Transition of a Carrier Detect (CD) input, transition of a Clear to Send (CTS) input, transition of sync input, entering or leaving synchronous Hunt Phase break detection or termination, SDLC abort detection or termination, Idle/CRC latch becoming set.

6.3.1.3.3 Control Register 2 - Channel A -- See Figure 6-11.



ADDRESS: 17775704, RP = 010, WRITE-ONLY

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	0	0	0	1	0	PRI	0	DMA MODE

Figure 6-11 Control Register 2 - Channel A

Bits	Name	Description
15:8		Not used (read as ones)
7:3		Must be 00010
2	PRI	Priority - If both channels A and B are in interrupt mode, the interrupt priority is RxA > TxA > RxB > TxB > extA > extB if PRI is cleared and RxA > RxB > TxA > TxB > extA > extB if PRI is set. If channel A is in DMA mode and channel B is in interrupt mode, the interrupt priority is RxA > RxB > TxB > extA > extB
1		Must be zero
0	DMA MODE	DMA mode - If set, channel A operates in DMA mode and channel B does not. If cleared, neither channel A nor B operates in DMA mode.

6.3.1.3.4 Control Register 2 - Channel B -- Control Register 2 for channel B holds the SLU2 interrupt vector (Figure 6-12). Although the register is programmed via channel B, the same vector is used for interrupts on both channels A and B. Initially, the KXJ11-CA firmware loads this vector with an octal value of 70. If bit 2 in Control Register 1 is set, the contents of this register will be modified according to the type of interrupt that occurs. The modified vector is obtained from Status Register 2 (see Section 6.3.1.4.3).

ADDRESS: 17775714, RP = 010, WRITE-ONLY

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	VECTOR							

MR-17210

Figure 6-12 Control Register 2 - Channel B

6.3.1.3.5 Control Register 3 -- See Figure 6-13.

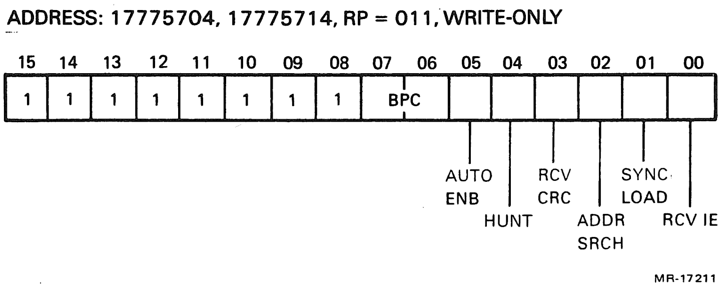


Figure 6-13 Control Register 3

Bits	Name	Description
15:8		Not used (read as ones)
7:6	BPC	Bits per character - Specifies the number of data bits per received character. BPC Bits Per Character 00 5 01 6 10 7 11 8
5	AUTO ENB	Auto enable - When set, causes Carrier Detect (CD) to act as an enable for the receiver and Clear to Send (CTS) to act as the enable for the transmitter.
4	HUNT	Hunt - When set, causes the receiver to enter a hunt phase. This is typically done to restore synchronization. When the receiver is enabled, a hunt begins and a transfer can occur only when character synchronization has been achieved. The hunt phase is also automatically entered whenever a channel is reset.
3	RCV CRC	Receiver CRC enable - When set, enables CRC calculation. When cleared, disables (but does not reset) the receiver CRC generator.
2	ADDR SRCH	Address search mode - This bit must be zero in non-SDLC modes. If this bit is set in SDLC mode, character assembly does not begin until the 8-bit character (secondary address field) following the starting flag of a message matches either the address programmed into Control Register 6 or the global address 11111111 (binary).

Bits	Name	Description
1	SYNC LOAD	Sync character load inhibit - When set, prevents the loading of sync characters into the receive buffer. Meaningful only in synchronous mode. When using CRC, this bit should be used to strip only the leading sync characters preceding a message and not the embedded sync characters. Protocols using other types of block checking, however, may use this bit to strip the embedded sync characters.
0	RCV IE	Receiver enable - When set, enables this channel's receiver. When cleared, disables the receiver.

6.3.1.3.6 Control Register 4 -- See Figure 6-14.

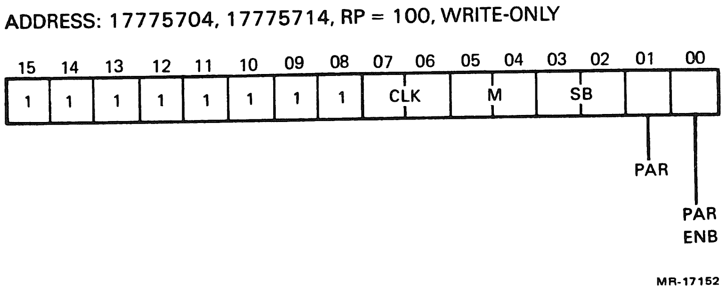


Figure 6-14 Control Register 4

Bits	Name	Description
15:8		Not used (read as ones)
7:6	CLK	Clock rate - Specify the relationship between the transmitter and receiver clock inputs and the actual data rate. When operating in synchronous mode, CLK must be 00.

CLK	Clock Rate
00	1 X data rate
01	16 X data rate
10	32 X data rate
11	64 X data rate

Bits Name Description

5:4 M

Sync Mode - Selects which synchronous protocol to use if this channel has been programmed in a synchronous mode.

M	Protocol
00	Monosynch
01	Bisynch
10	SDLC
11	External Synch

3:2 SB

Stop bits/synchronous mode - Specifies whether the channel will be used in synchronous or asynchronous mode. In asynchronous mode, this field also specifies the number of stop bits used by the transmitter. The receiver always uses one stop bit.

SB	Mode
00	Synchronous mode
01	Asynchronous mode, 1 stop bit
10	Asynchronous mode, 1.5 stop bits
11	Asynchronous mode, 2 stop bits

1 PAR

Parity sense - When set, causes even parity generation and checking. When cleared, causes odd parity generation and checking.

0 PAR ENB

Parity enable - When set, causes an extra bit containing parity information to be concatenated with each transmitted character. Also causes parity checking to be performed for each received character.

6.3.1.3.7 Control Register 5 -- See Figure 6-15.

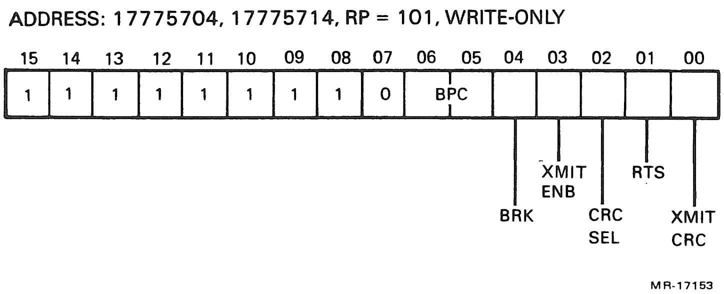


Figure 6-15 Control Register 5

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th></tr>
<tr><td>15:8</td><td></td><td>Not used (read as ones)</td></tr>
<tr><td>7</td><td></td><td>Must be zero</td></tr>
<tr><td>6:5</td><td>BPC</td><td>Transmitted bits per character - Specifies the number of data bits per transmitted character.
<table>
<tr><th>BPC</th><th>Bits/Character</th></tr>
<tr><td>00</td><td>5 or less</td></tr>
<tr><td>01</td><td>7</td></tr>
<tr><td>10</td><td>6</td></tr>
<tr><td>11</td><td>8</td></tr>
</table>
Note that for five or less bits/character, the data must be formatted as follows:
<table>
<tr><th>Bits/Character</th><th>Format</th></tr>
<tr><td>1</td><td>1111000d</td></tr>
<tr><td>2</td><td>111000dd</td></tr>
<tr><td>3</td><td>11000ddd</td></tr>
<tr><td>4</td><td>1000dddd</td></tr>
<tr><td>5</td><td>000ddddd</td></tr>
</table>
Where d represents a data bit. The most significant data bit is always in the leftmost position.</td></tr>
<tr><td>4</td><td>BRK</td><td>Send break - When set, forces this channel's transmitter data output low (spacing).</td></tr>
<tr><td>3</td><td>XMIT ENB</td><td>Transmitter enable - When this channel is reset, this bit is cleared. The transmitter data output is forced high (marking) and the transmitter is disabled until this bit is set.</td></tr>
<tr><td>2</td><td>CRC SEL</td><td>CRC polynomial select - When set, the CRC-16 polynomial is selected (X**16 + X**15 + X**2 + 1). When cleared, the CRC-CCITT polynomial is selected (X**16 + X**12 + X**5 + 1). The CRC-CCITT polynomial must be selected when in SDLC mode.</td></tr>
<tr><td>1</td><td>RTS</td><td>Request to Send - When set, asserts RTS. When cleared, deasserts RTS. In synchronous and SDLC modes, RTS is asserted immediately. In asynchronous mode, RTS is asserted only when the transmitter data buffer is completely empty.</td></tr>
</table>

Bits	Name	Description
0	XMIT CRC	Transmitter CRC enable - When set, enables this channel's transmitter CRC generator. When cleared, the CRC calculation is not performed. Setting and clearing this bit includes or excludes individual characters from a CRC calculation. If this bit is cleared when a transmitter underrun occurs, the CRC will not be sent.

6.3.1.3.8 Control Register 6 -- Control Register 6 (Figure 6-16) holds sync byte 1, which has different meanings in different modes.

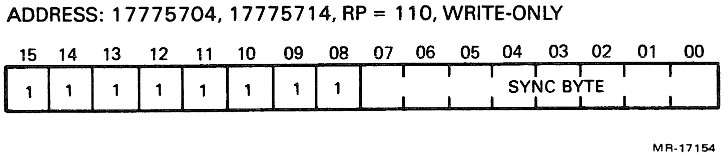


Figure 6-16 Control Register 6

- Monosync - The 8-bit sync character transmitted during the Idle phase.
- Bisync - The least significant 8 bits of the 16-bit transmit and receive sync character.
- SDLC - A secondary address value that is matched to the Secondary Address field of the SDLC frame when in Address Search mode.
- External Sync - The sync character transmitted during the Idle phase.

6.3.1.3.9 Control Register 7 -- Control Register 7 (Figure 6-17) holds sync byte 2, which has different meanings in different modes:

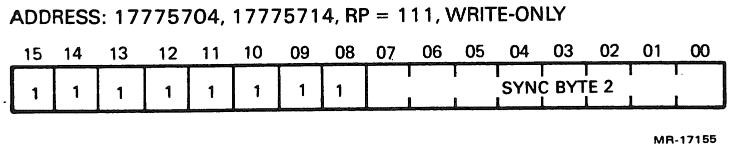


Figure 6-17 Control Register 7

- Monosync - The 8-bit sync character matched by the receiver.

- Bisync - The most significant 8 bits of the 16-bit transmit and receive sync character.

- SDLC - Must contain the flag character (01111110) matched by the receiver.

- External Sync - Control Register 7 is not used in external sync mode.

6.3.1.4 SLU2 Status Registers -- Channel A has two read-only Status Registers numbered 0 and 1. Channel B has three read-only Status Registers numbered 0 through 2. A Status Register is read by first writing Control Register 0 with an appropriate register pointer. Then, a read to address 17775700 (for channel A) or 17775710 (for channel B) produces the status data. This section describes each of the Status Registers.

6.3.1.4.1 Status Register 0 -- See Figure 6-18.

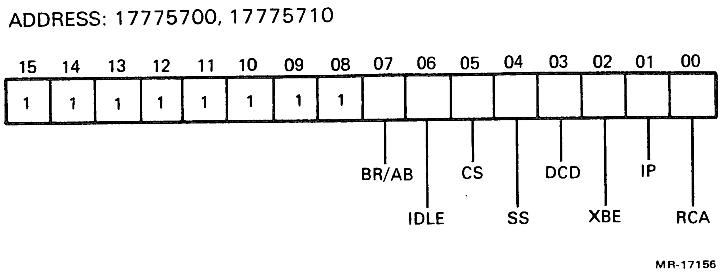


Figure 6-18 Status Register 0

Bits	Name	Description
15:8		Not used (read as ones)
7	BR/AB	Break/Abort - When set in asynchronous receive mode, indicates that a break sequence has been detected. A break occurs when the data input is held low (spacing) for more than one character time. Cleared when the input returns high (marking). An External/Status interrupt, if enabled, occurs when the state of this bit changes. When set in SDLC mode, indicates that an abort sequence (seven or more 1's) has been detected.
6	IDLE	Idle - Indicates the state of the Idle/CRC latch used in synchronous and SDLC modes. This bit is set during a reset operation, and cleared by a Reset Transmit Underrun/EOM Latch command.

Bits	Name	Description
5	CS	Clear to send - This bit reflects the state of the CTS input for this channel. When set, CTS is asserted. Any transition of this bit causes an External/Status interrupt request.
4	SS	Sync status - Meaning depends upon the operating mode of this channel. Asynchronous: Reflects the state of the SYNC input. When set, SYNC is asserted. Any transition of this bit causes an External/Status interrupt request. External sync mode: Similar to asynchronous mode. A low-to-high transition of this bit indicates that synchronization has been achieved and character assembly has begun. Monosync, Bisync, SDLC modes: When set, indicates that the receiver is in the Sync Hunt phase of operation. When cleared, indicates that the receiver is in the Receive Data phase.
3	DCD	Data carrier detect - Reflects the state of the DCD input. When set, DCD is asserted. Any transition of this bit causes an External/Status interrupt request.
2	XBE	Transmitter buffer empty - When set, indicates that the transmitter buffer is empty, except during transmission of CRC characters in synchronous mode. When cleared, indicates the transmitter buffer is loaded. This bit is set during a reset operation.

Bits Name Description

1 IP Interrupt pending (Channel A only) - Used in conjunction with the interrupt vector register (Status Register 2, channel B) to determine the status of an SLU2 interrupt. In Non-Vectored Interrupt mode, this bit is set when Status Register 2, channel B is read. The low three bits of this Status Register indicate the cause of the interrupt. In Vectored Interrupt mode, the Interrupt Pending bit is set when SLU2 is the highest priority device requesting interrupt service. In either mode, the bit is cleared when an End of Interrupt command is issued and there are no other pending interrupt requests. This bit is zero for channel B.

0 RCA Received character enable - When set, indicates that one or more characters are available in the receiver buffer. Once all the available characters have been read, this bit is cleared until a new character has been received.

6.3.1.4.2 Status Register 1 -- See Figure 6-19.

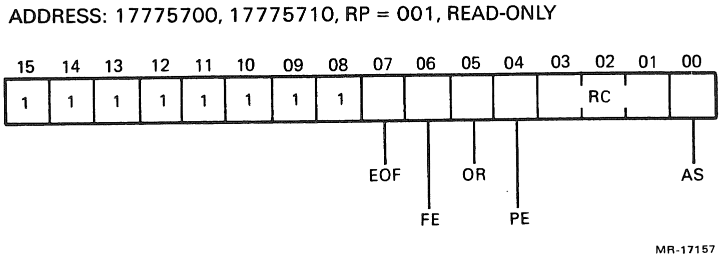


Figure 6-19 Status Register 1

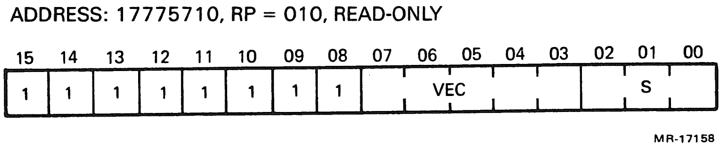
Bits Name Description

15:8 Not used (read as ones)

7 EOF End of frame - This bit is valid only in SDLC mode. When set, indicates that a valid ending flag has been received and that the CRC error flag and residue code are valid. Cleared by an Error Reset command or upon reception of the first character of the next frame.

Bits	Name	Description
6	FE	Framing error - When set in asynchronous mode, indicates that no stop bit has been detected at the end of a received character. When set in synchronous modes, indicates that the calculated CRC value does not match the last two bytes received. This bit is cleared by issuing an Error Reset command.
5	OR	Overrun error - When set, indicates that the receiver buffer has been overloaded. The receiver buffer (FIFO) can contain three characters. If a fourth character is received, the last character in the buffer is overwritten. This error bit remains latched until an Error Reset command is issued.
4	PE	Parity error - When set, indicates that parity checking has been enabled and that the parity of a received character matches the programmed sense (even/odd). This bit remains set until an Error Reset command is issued.
3:1	RC	SDLC residue code - These bits are valid only in SDLC mode. The data portion of an SDLC message may consist of a non-integral number of characters. Since transfers are character oriented, the residue code provides the capability to receive any leftover bits. See the uPD7201 Data Sheet for a table of residue codes corresponding to characters of various lengths.
0	AS	All sent - When set in asynchronous mode, indicates that the transmitter buffer is empty. When cleared in asynchronous mode, indicates that a character is present in the transmitter buffer or shift register. In synchronous modes, this bit is always set.

6.3.1.4.3 Status Register 2 (Channel B Only) -- See Figure 6-20.



ADDRESS: 17775710, RP = 010, READ-ONLY

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
1	1	1	1	1	1	1	1	VEC					S		

MR-17158

Figure 6-20   Status Register 2 (Channel B Only)

Bits	Name	Description
15:8		Not used (read as ones)
7:3	VEC	Interrupt vector - Contains bits <7:3> of the vector contained in Control Register 2, channel B (see Section 6.3.1.3.4).
2:0	S	Status modifiers - These three bits indicate the following:

S	Description
111	No Interrupt Pending
000	Channel B Transmitter Buffer Empty
001	Channel B External/Status Change
010	Channel B Received Character Available
011	Channel B Special Receive Condition
100	Channel A Transmitter Buffer Empty
101	Channel A External/Status Change
110	Channel A Received Character Available
111	Channel A Special Receive Condition

111 has two meanings. They may be distinguished by examining bit 1 of Status Register 0, channel A (Interrupt Pending).

6.3.1.5 SLU2 Transmitter Registers -- There are two Transmitter Data Registers, one for each channel. The format of these registers is shown in Figure 6-21.

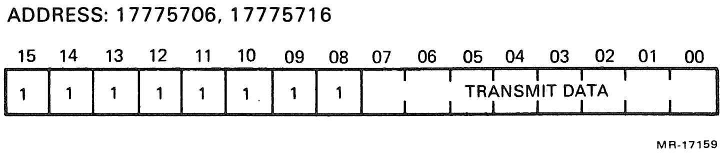


Figure 6-21 Transmitter Registers A and B

6.3.1.6 SLU2 Receiver Registers -- There are two Receiver Registers (data registers), one for each channel. The format of these registers is shown in Figure 6-22.

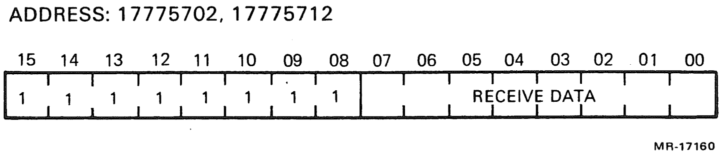


Figure 6-22 Receiver Registers A and B

6.3.2 Examples

The following programs provide "skeletons" on which to base user application programs.

```
.TITLE  SLU1.MAC

;  This program utilizes the uPD7201 to transfer serial data. The
;  data will be transferred out of Channel A and received by
;  Channel A, so a loopback connector is required (Part #H3022 or
;  54-16229-01). This example transfers the data in asynchronous
;  mode using interrupts.
;
;  After this program has been assembled and linked on the
;  development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the program into the KXJ11-CA. This
;  program will execute as shown in the following example.
;
```

```
;  SET 2
;  LOAD SLU1.SAV
;  EXECUTE
;  !ODT
;  !
;  !001206
;  !001302/041101
;  !001304/042103
;  !001306/043105
;  !001310/044107
;  !001312/041101
;  !001314/042103
;  !001316/043105
;  !001320/044107
;  !001322/000000
;  !R4/000000
;  !CTRL/C
;  EXIT
;
;  This verifies that the data was successfully transferred. 1302
;  is the address of the transmit buffer and 1312 is the address
;  of the receive buffer. R4=0 verifies that no external or
;  special condition interrupts were received.
;
;  Register Definitions

     STATA   ==  175700       ; Channel A status register
     RBUFA   ==  175702       ; Channel A receiver
     CNTRLA  ==  175704       ; Channel A control register
     TBUFA   ==  175706       ; Channel A transmitter

     STATB   ==  175710       ; Channel B status register
     CNTRLB  ==  175714       ; Channel B control register

     TIMREG  ==  175736       ; Timer control register
     TIMER0  ==  175730       ; Timer 0 data register

START::

;  This section initializes the KXJ11-CA system environment

     MTPS    #340             ; Disable recognition of interrupts

     MOV     #ISR,@#70        ; SLU2 interrupts at location 70
     MOV     #340,@#72        ; Let the ISR run at priority 7

     CLR     R0               ; This is the transmit char counter

     MOV     #TBUF,R2         ; R2 points to the transmit buffer
     MOV     #RBUF,R3         ; R3 points to the receive buffer

     CLR     R4               ; This counter keeps track of
                              ; external status changes and
                              ; special receive conditions
;   This section initializes the bit rate generator

        MOVB    #26,TIMREG      ; Select timer 0, low byte only,
                                ; mode 3, binary
        MOVB    #64.,TIMER0     ; This divider selects 9600 bps

;   This section initializes the 7201 for asynch operation

        MOVB    #30,CNTRLA      ; Reset Channel A
        NOP                     ; Wait for reset to complete

        MOVB    #30,CNTRLB      ; Reset Channel B
        NOP                     ; Wait for reset to complete

        MOVB    #2,CNTRLA       ; Point to CR2A
        MOVB    #24,CNTRLA      ; Setup bus interface options:
                                ; No DMA, RxA>RxB>TxA...,
                                ; Non-Vectored

        MOVB    #4,CNTRLA       ; Point to CR4
        MOVB    #104,CNTRLA     ; Set operation mode:
                                ; No parity, asynch mode, 1 stop bit,
                                ; clock rate = 16x data rate

        MOVB    #3,CNTRLA       ; Point to CR3
        MOVB    #301,CNTRLA     ; Enable receiver, char length = 8

        MOVB    #5,CNTRLA       ; Point to CR5
        MOVB    #152,CNTRLA     ; Enable transmitter, Char length = 8

        CLRB    CNTRLA          ; Point to CR0
        MOVB    #20,CNTRLA      ; Reset External/Status Interrupts

        MOVB    #1,CNTRLA       ; Point to CR1
        MOVB    #36,CNTRLA      ; Transmit IE, Interrupt on all
                                ; received chars, enable condition
                                ; affects vector

MAIN::
        MTPS    #0              ; Enable recognition of interrupts
        MOVB    (R2)+,TBUFA     ; Send first character
        BR      .               ; Stay here while the interrupts
                                ; occur
```

```
ISR::
     MOVB    #2,CNTRLB           ; Point to SR2B
     MOVB    STATB,-(SP)         ; Store the condition affects vector
                                 ; on the stack

     ;  This section inspects the Condition Affects vector to
     ;  determine the cause of the interrupt

     ROR     (SP)                ; Rotate bit 0 into the carry bit
     BCS     EXT                 ; If this bit was set then the
                                 ; interrupt was caused by a special
                                 ; receive condition or an external/
                                 ; status change
     ROR     (SP)                ; Rotate bit 1 into the carry bit
     BCS     RCV                 ; If this bit was set then the
                                 ; interrupt was caused by a received
                                 ; character
     ;+
     ;  If neither of the above conditions was
     ;  satisfied then the interrupt must have
     ;  been caused by the transmitter buffer
     ;  going empty
     ;-
XMIT::
     INC     R0                  ; Increment the xmit char counter
     CMP     R0,#8.              ; IF this is the eight char
     BEQ     1$                  ; THEN branch to 1$
     MOVB    (R2)+,TBUFA         ; ELSE send another char
     BR      IDONE               ; and return
1$:  MOVB    #50,CNTRLA          ; reset pending xmit interrupt
     BR      IDONE               ; request - then return
RCV:: MOVB   RBUFA,(R3)+         ; Store this character
     BR      IDONE               ; and return

EXT::;  This program does not take any special action if an
     ;  External/Status interrupt or Special Receive Condition
     ;  occurs.  Just note that it occurred (there shouldn't be
     ;  any) and continue.

     INC     R4                  ; Increment the counter
                                 ; and return

IDONE:: TST  (SP)+               ; Fix the stack
     MOVB    #70,CNTRLA          ; Issue end of interrupt command
     RTI                         ; and return to main program

TBUF:: .BYTE   101,102,103,104,105,106,107,110
RBUF:: .BLKB   8.

.END    START
```

```
.TITLE  SLU2.MAC

;  This example program for the uPD7201 transfers serial data via
;  a loopback connector (part #H3022 or 54-16229) between Channel
;  A's transmit and receive, using the DMA controller.  No ISR is
;  included in this example as it is meant to show how the uPD7201
;  and the DTC may work together.  A "real-life" program should
;  include an ISR which monitors any External or Special Receive
;  condition interrupts.  For more information regarding the
;  programming of the DTC please refer to Section 4.3.
;
;  After this program has been assembled and linked on the
;  development machine, use the KUI utility of the KXJ11-CA
;  Software Toolkit to load the program into the KXJ11-CA. This
;  program will execute as shown in the following example.
;
;  SET 2
;  LOAD SLU2.SAV
;  EXECUTE
;  !ODT
;  !
;  !001234
;  !1276/041101
;  !001300/042103
;  !001302/043105
;  !001304/044107
;  !001306/041101
;  !001310/042103
;  !001312/043105
;  !001314/044107
;  !001316/000000
;  !CTRL/C
;  EXIT
;  This verifies that the data was transferred successfully. The
;  transmit buffer begins at address 1276, and the receive buffer
;  begins at address 1306.
;
;  Register Assignments

     MMREG    ==   174470      ; Master Mode Register
     CMDREG   ==   174454      ; Command Register
     CASTF0   ==   174446      ; Chan 0 Chain Address Seg/Tag Field
     CAOF0    ==   174442      ; Chan 0 Chain Address Offset Field
     CASTF1   ==   174444      ; Chan 1 Chain Address Seg/Tag Field
     CAOF1    ==   174440      ; Chan 1 Chain Address Offset Field

     STATA    ==   175700      ; Channel A status register
     RBUFA    ==   175702      ; Channel A receiver
     CNTRLA   ==   175704      ; Channel A control register
     TBUFA    ==   175706      ; Channel A transmitter
     STATB    ==   175710      ; Channel B status register
     CNTRLB   ==   175714      ; Channel B control register
     TIMREG   ==   175736      ; Timer control register
     TIMER0   ==   175730      ; Timer 0 data register
```

```
START::

;  This section initializes the KXJ11-CA system environment

     MTPS    #340              ; Disable recognition of interrupts

     MOV     #TBUF,R2          ; R2 points to the transmit buffer
     MOV     #RBUF,R3          ; R3 points to the receive buffer

;  This section initializes the bit rate generator

     MOVB    #26,TIMREG        ; Select timer 0, low byte only,
                               ; mode 3, binary
     MOVB    #64.,TIMER0       ; This divider selects 9600 bps

;  This section initializes the 7201 for asynch operation

     MOVB    #30,CNTRLA        ; Reset Channel A
     NOP                       ; Wait for reset to complete

     MOVB    #30,CNTRLB        ; Reset Channel B
     NOP                       ; Wait for reset to complete

     MOVB    #2,CNTRLA         ; Point to CR2A
     MOVB    #25,CNTRLA        ; Setup bus interface options:
                               ; Chan A DMA, RxA>RxB>TxA...,
                               ; Non-Vectored

     MOVB    #4,CNTRLA         ; Point to CR4
     MOVB    #104,CNTRLA       ; Set operation mode:
                               ; No parity, asynch mode, 1 stop bit,
                               ; clock rate = 16x data rate

     MOVB    #3,CNTRLA         ; Point to CR3
     MOVB    #301,CNTRLA       ; Enable receiver, char length = 8
                               ;
     MOVB    #5,CNTRLA         ; Point to CR5
     MOVB    #152,CNTRLA       ; Enable transmitter, Char length = 8
                               ;
     CLRB    CNTRLA            ; Point to CR0
     MOVB    #20,CNTRLA        ; Reset External/Status Interrupts
                               ;
     MOVB    #1,CNTRLA         ; Point to CR1
     MOVB    #16,CNTRLA        ; Transmit IE, Interrupt on 1st
                               ; received char and issue DMA
                               ; request enable condition affects
                               ; vector
```

```
   This section initializes the DMA controller

     CLRB    CMDREG            ; Reset the DTC

     MOV     #0,CASTF0         ; Load Chain Address Register Seg/Tag
     MOV     #LOAD0,CAOF0      ; Load Chain Address Register Offset
     MOV     #0,CASTF1         ; Load Chain Address Register Seg/Tag
     MOV     #LOAD1,CAOF1      ; Load Chain Address Register Offset

     MOVB    #115,MMREG        ; Load Master Mode Reg to Enable DTC

     MOVB    #240,CMDREG       ; Start Chain Channel 0
     MOVB    #241,CMDREG       ; Start Chain Channel 1

MAIN::
     BR      .                 ; Stay here while the DMA transfers
                               ; occur

     ; Chain Load Region

     LOAD1: .WORD   001602     ; Reload Word <Select
                               ; CARA,CARB,COPC,CM>

     .WORD  000000   ; Current Address Register A Seg/Tag
     .WORD  TBUF     ; Current Address Register A Offset
                     ; <This local address is the source>

     .WORD  000020   ; Current Address Register B Seg/Tag
     .WORD  TBUFA+1  ; Current Address Register B Offset
                     ; <This local address is the destination>

     .WORD  000010   ; Current Operation Count <Transfer 8 bytes>

     .WORD  000020   ; Channel Mode Register High
     .WORD  000001   ; Channel Mode Register Low
                     ; <No match conditions, do nothing upon
                     ; completion, transfer type = single transfer
                     ; CARA = source, byte transfers>
```

```
LOAD0::.WORD 001602  ; Reload Word <Select CARA,CARB,COPC,CM>
     .WORD  000020   ; Current Address Register A Seg/Tag
     .WORD  RBUFA+1  ; Current Address Register A Offset
                     ; <This local address is the source>

     .WORD  000000   ; Current Address Register B Seg/Tag
     .WORD  RBUF     ; Current Address Register B Offset
                     ; <This local address is the destination>

     .WORD  000010   ; Current Operation Count <Transfer 8 bytes>

     .WORD  000000   ; Channel Mode Register High
     .WORD  000001   ; Channel Mode Register Low

                     ; <No match conditions, do nothing upon
                     ; completion, transfer type = single transfer
                     ; CARA = source, byte transfers>

TBUF:: .BYTE         101,102,103,104,105,106,107,110
RBUF:: .BLKB   10

.END    START
```

APPENDIX A
MEMORY MAP SUMMARY

A.1 REGISTER SUMMARY

Table A-1 lists all the registers in the KXJ11-CA and specifies the addresses associated with these registers.

Table A-1 KXJ11-CA Registers

KXJ11-CA Address	Register
17772200-17772216	Supervisor I Space PDR0-PDR7
17772220-17772236	Supervisor D Space PDR0-PDR7
17772240-17772256	Supervisor I Space PAR0-PAR7
17772260-17772276	Supervisor D Space PAR0-PAR7
17772300-17772316	Kernel I Space PDR0-PDR7
17772320-17772336	Kernel D Space PDR0-PDR7
17772340-17772356	Kernel I Space PAR0-PAR7
17772360-17772376	Kernel D Space PAR0-PAR7
17772516	Memory Management Register 0 (MMR3)
17774400	DTC CH1 Current B Address Offset
17774402	DTC CH0 Current B Address Offset
17774404	DTC CH1 Base B Address Offset
17774406	DTC CH0 Base B Address Offset
17774410	DTC CH1 Current A Address Offset
17774412	DTC CH0 Current A Address Offset
17774414	DTC CH1 Base A Address Offset
17774416	DTC CH0 Base A Address Offset
17774420	DTC CH1 Current B Address Segment/Tag
17774422	DTC CH0 Current B Address Segment/Tag
17774424	DTC CH1 Base B Address Segment/Tag
17774426	DTC CH0 Base B Address Segment/Tag
17774430	DTC CH1 Current A Address Segment/Tag
17774432	DTC CH0 Current A Address Segment/Tag
17774434	DTC CH1 Base A Address Segment/Tag
17774436	DTC CH0 Base A Address Segment/Tag
17774440	DTC CH1 Chain Address Offset
17774442	DTC CH0 Chain Address Offset
17774444	DTC CH1 Chain Address Segment/Tag
17774446	DTC CH0 Chain Address Segment/Tag
17774450	DTC CH1 Interrupt Save Register
17774452	DTC CH0 Interrupt Save Register

Table A-1 KXJ11-CA Registers (Cont)

KXJ11-CA Address	Register
17774454	DTC CH1 Status Register (Read Only)
17774454	DTC Command Register (Write Only)
17774456	DTC CH0 Status Register
17774460	DTC CH1 Current Operation Count
17774462	DTC CH0 Current Operation Count
17774464	DTC CH1 Base Operation Count
17774466	DTC CH0 Base Operation Count
17774470	DTC Master Mode Register
17774472-1774506	DTC Reserved
17774510	DTC CH1 Pattern Register
17774512	DTC CH0 Pattern Register
17774514	DTC CH1 Mask Register
17774516	DTC CH0 Mask Register
17774520	DTC CH1 Channel Mode Low
17774522	DTC CH0 Channel Mode Low
17774524	DTC CH1 Channel Mode High
17774526	DTC CH0 Channel Mode High
17774530	DTC CH1 Interrupt Vector
17774532	DTC CH0 Interrupt Vector
17774534-17774536	DTC Reserved
17775000	TPR0
17775002	TPR1
17775004	TPR2
17775006	TPR3
17775010	TPR4
17775012	TPR5
17775014	TPR6
17775016	TPR7
17775020	TPR8
17775022	TPR9
17775024	TPR10
17775026	TPR11
17775030	TPR12
17775032	TPR13
17775034	TPR14
17775036	TPR15
17775700	SLU2 Channel A Status Register
17775702	SLU2 Channel A Receiver
17775703	SLU2 Channel A Receiver (when used with DTC)
17775704	SLU2 Channel A Control Register
17775706	SLU2 Channel A Transmitter
17775707	SLU2 Channel A Transmitter (when used with DTC)
17775710	SLU2 Channel B Status Register
17775712	SLU2 Channel B Receiver
17775714	SLU2 Channel B Control Register
17775716	SLU2 Channel B Transmitter
17775720	SLU2 Timer 0 Data Register

Table A-1 KXJ11-CA Registers (Cont)

KXJ11-CA Address	Register
17775722	SLU2 Timer 1 Data Register
17775724	SLU2 Timer 2 Data Register
17775730	SLU2 Timer 0 Data Register
17775732	SLU2 Timer 1 Data Register
17775734	SLU2 Timer 2 Data Register
17775736	SLU2 Timer Control Register
17777000	PIO Master Interrupt Control Register
17777002	PIO Master Configuration Control Register
17777004	PIO Port A Interrupt Vector Register
17777006	PIO Port B Interrupt Vector Register
17777010	PIO Counter/Timer Interrupt Vector Register
17777012	PIO Port C Data Path Polarity Register
17777014	PIO Port C Data Direction Register
17777016	PIO Port C Special I/O Control Register
17777020	PIO Port A Command and Status Register
17777022	PIO Port B Command and Status Register
17777024	PIO Counter/Timer 1 Command and Status Register
17777026	PIO Counter/Timer 2 Command and Status Register
17777030	PIO Counter/Timer 3 Command and Status Register
17777032	PIO Port A Data Register
17777033	PIO Port A Data Register (when used with DTC)
17777034	PIO Port B Data Register
17777035	PIO Port B Data Register (when used with DTC)
17777036	PIO Port C Data Register
17777040	PIO Counter/Timer 1 Current Count (MSB)
17777042	PIO Counter/Timer 1 Current Count (LSB)
17777044	PIO Counter/Timer 2 Current Count (MSB)
17777046	PIO Counter/Timer 2 Current Count (LSB)
17777050	PIO Counter/Timer 3 Current Count (MSB)
17777052	PIO Counter/Timer 3 Current Count (LSB)
17777054	PIO Counter/Timer 1 Time Constant (MSB)
17777056	PIO Counter/Timer 1 Time Constant (LSB)
17777060	PIO Counter/Timer 2 Time Constant (MSB)
17777062	PIO Counter/Timer 2 Time Constant (LSB)
17777064	PIO Counter/Timer 3 Time Constant (MSB)
17777066	PIO Counter/Timer 3 Time Constant (LSB)
17777070	PIO Counter/Timer 1 Mode Specification
17777072	PIO Counter/Timer 2 Mode Specification
17777074	PIO Counter/Timer 3 Mode Specification
17777076	PIO Current Vector Register
17777100	PIO Port A Mode Specification Register
17777102	PIO Port A Handshake Specification Register
17777104	PIO Port A Data Path Polarity Register
17777106	PIO Port A Data Direction Register
17777110	PIO Port A Special I/O Control Register
17777112	PIO Port A Pattern Polarity Register (PPR)
17777114	PIO Port A Pattern Transition Register (PTR)
17777116	PIO Port A Pattern Mask Register (PMR)
17777120	PIO Port B Mode Specification Register

Table A-1 KXJ11-CA Registers (Cont)

KXJ11-CA Address	Register
17777122	PIO Port B Handshake Specification Register
17777124	PIO Port B Data Path Polarity Registers
17777126	PIO Port B Data Direction Registers
17777130	PIO Port B Special I/O Control Registers
17777132	PIO Port B Pattern Polarity Registers (PPR)
17777134	PIO Port B Pattern Transition Registers (PTR)
17777136	PIO Port B Pattern Mask Register (PMR)
17777140	PIO I/O Buffer Control Register
17777520	KXJ11 Control/Status Register A (KXJCSRA)
17777522	KXJ11 Control/Status Register B (KXJCSRB)
17777524	KXJ11 Control/Status Register C (KXJCSRC)
17777526	KXJ11 Control/Status Register E (KXJCSRE)
17777530	KXJ11 Control/Status Register D (KXJCSRD)
17777532	Q-Bus Interrupt Register (QIR)
17777534	KXJ11 Control/Status Register F (KXJCSRF)
17777536	KXJ11 Control/Status Register H (KXJCSRH)
17777540	KXJ11 Control/Status Register J (KXJCSRJ)
17777560	SLU1 Receiver Control/Status Register (RCSR)
17777562	SLU1 Receiver Buffer Register (RBUF)
17777564	SLU1 Transmitter Control/Status Register (XCSR)
17777566	SLU1 Transmitter Buffer Register (XBUF)
17777572	Memory Management Register 0 (MMR0)
17777574	Memory Management Register 1 (MMR1)
17777576	Memory Management Register 2 (MMR2)
17777600-17777616	User I Space PDR0-PDR7
17777620-17777636	User D Space PDR0-PDR7
17777640-17777656	User I Space PAR0-PAR7
17777660-17777676	User D Space PAR0-PAR7
17777750	Maintenance Register
17777766	CPU Error Register
17777772	PIR
17777776	Processor Status Word (PSW)

APPENDIX B
KXJ11-CA/KXT11-CA DIFFERENCES

B.1 DIFFERENCES BETWEEN THE KXJ11-CA AND THE KXT11-CA

Table B-1 summarizes the differences between the KXJ11-CA and the KXT11-CA.

Table B-1 KXJ11-CA/KXT11-CA Differences

	KXJ11-CA	KXT11-CA
Memory management	Yes	No
PROM	64 kilobytes	8-32 kilobytes
RAM	512 kilobytes	32-48 kilobytes
RAM parity	Yes	No
Shared memory	Yes	No
Warm floating-point	Yes	No
Maintenance Register	Yes	No
DTC vectors	214 and 220	110 and 114
TPR0 (control mode)		
TPR0<14>	Hard reset	Unused
TPR0<9>	Execute program	Unused
TPR0<8>	Disable shared memory	Unused
TPR0<7>	Show shared memory	Unused
TPR0<6>	Enable shared memory	Unused
TPR1		
TPR1<13>	Unused	Trap to 4 disable
TPR1<12>	Non-existent memory error	Unused
TPR1<11>	Parity error	SP NXM test flag
TPR1<10>	Unused	Power-up with battery backup

Table B-1 KXJ11-CA/KXT11-CA Differences (Cont)

	KXJ11-CA	KXT11-CA
TPR1<9>	Unused	Power-up without battery backup
TPR1<7>	Unused	Q-Bus ODT flag
TPR1<6>	Unused	Serial ODT flag
TPR1<4>	Firmware not handling special interrupts	ODT on HALT instruction
TPR1<3>	Unused	Stack error flag
CSRB		
CSRB<7:4>	Writeable	Read-only
CSRB<3:1>	Base address and bus size jumpers	Memory mapping jumpers
CSRD<15>	PWR FL	NXM
CSRE	No operation	Controls PIO
CSRF	Yes	No
CSRH	Yes	No
HALT instruction	If in kernel mode, enters serial ODT	Restart
Stack violations and NXM references in kernel mode	Fatal runtime error	Not applicable
Exceptions	Caused by the assertion of BHALT or BINIT, the deassertion of BPOK, or the writing of TPR0 by the arbiter	HALT instruction
Battery backup	No	Yes
Firmware stack	128 kilobytes at top of kernel stack for native firmware scratch area	Separate RAM in I/O page - transparent to user
Hardware reset	Caused by power-up or by setting TPR0<14>	Caused by power-up
Arbiter NOP command	Ignored	Reserved

Table B-1 KXJ11-CA/KXT11-CA Differences (Cont)

	KXJ11-CA	KXT11-CA
Boot/self-test switch	Switch positions 0-6 in KXJ11-CA are identical to switch positions 0-6 in KXT11-CA. Switch positions 7-15 are unique.	
ID Switch	Switch position changes made while the board is powered up take effect only after a hardware reset.	Switch position changes made while the board is powered up immediately affect Q-Bus address.
Console ODT	Microcode	Firmware
	All alpha characters must be upper case	R and S may be upper or lower case
	^ (close last memory location or register and open preceding) not supported	Supported
	Examining a range of locations not supported	Supported
	Register identifier can be preceded by a $ as well as an R	Not supported
	Autobaud not supported	Supported
	All addresses are 22 bits	All addresses are 16 bits
	Two register sets and three stack pointers	Not applicable
	LEDs Indeterminate	LEDs in fixed state while in ODT
Software control of SLU1 baud rate	Selectable by a bit in KXJCSRJ	Jumper selectable

APPENDIX C
USER (P)ROM PROGRAMMING INFORMATION

C.1 INTRODUCTION

This appendix contains the guidelines for installing user applications in PROM or ROM.

There are two sockets for PROM on the KXJ11-CA, E22 and E23. E23 contains the high byte of a PDP-11 word, and E22 contains the low byte. The KXJ11-CA is supplied with two 8 X 8K PROMS to provide 16 kilobyte of PROM space. The KXJ11-CA can also accommodate 8 X 16K, or 8 X 32K PROMs in these sockets to provide 32 KB and 64 KB of PROM space respectively. The PROMs supplied with the KXJ11-CA contain the native firmware code. The native firmware consists of initialization code, which is executed when power is first applied. Under certain reset conditions, the native firmware contains routines that interpret commands received from TPR0 and handle non-maskable interrupts and self-test programs.

Since these are the only PROM sites on the module, user applications in PROM must reside in the same PROM as the native firmware.

The native firmware requires 16 kilobytes of PROM. Either of the two larger parts may be selected to provide 16 kilobytes or 48 kilobytes of user (P)ROM.

C.2 TRANSPORTING NATIVE FIRMWARE

If users want to replace the PROMs supplied with the KXJ11-CA with PROMs of a larger size, and also retain the native firmware functions, then they must transfer the firmware code in the Digital-supplied PROMs to the new PROMs.

The requirements and procedure for doing this are as follows:

C.2.1 Requirements

C.2.1.1 Hardware -- Any VAX-11 system supported by the VAX/VMS operating system, except VAX-11/730 dual RL02 configurations.

A PROM programmer that adheres to the command and data transfer protocol used by VAX DECprom. Included are:

DATA I/O Model 19
DATA I/O Model 29A
DATA I/O Model 121A
DATA I/O Model 171

Any necessary hardware, such as programming modules and socket adapters, specific to the PROMs being used.

A spare, dedicated VAX/VMS terminal port and cable to connect to and communicate with the PROM programmer.

C.2.1.2 Software -- Version 3.2 or later of the VAX/VMS operating system. VAX DECprom Utility

See the VAX DECprom User's Guide, Chapter 2, for further details.

C.2.2 Procedure

Remove the PROMs that are supplied with the KXJ11-CA from their sockets.

C.2.2.1 Setting Parameters -- DECprom requires PROM parameters, such as PROM length, word width, unprogrammed state, and the pinout code, to be set whenever data is to be transferred to or from PROM. This is done using the PARAMETERS command.

Invoking the PARAMETERS command caused the following display to be printed on the terminal:

```
CURRENT PARAMETER VALUES

PROM Parameters
1) FAMILY AND PINOUT CODE (4 HEX characters) ...............
2) LENGTH (power of 2, 2 -> 1073741824) ..................... 16384
3) WIDTH (1 -> 8) ................................................ 8
4) UNPROGRAMMED STATE (0/1) ...................................... 1
5) COLUMN POSITION IN PROM SET (0 through WIDTH-1) ............. 0
6) PROGRAMMER RAM BUFFER SIZE (power of 2, 2 -> 65536) ..... 16384

TARGET Parameters
7) WORD SIZE IN BITS (1 -> 128) ................................ 16
8) NUMBER OF BITS PER ADDRESS (1 -> word size) ................. 8
9) INVERTED DATA (YES/NO) ...................................... NO
10) INVERTED ADDRESS (YES/NO) .................................. NO
```

```
TYPE THE NUMBER TO CHANGE PARAMETER (1/2/3..../EXIT) <EXIT>:
```

The above parameters would be correct to read the contents of the native firmware in the 8 x 8KB PROM except for item 1. To change the parameter of item 1; type the number 1 at the colon. The terminal will display the following:

```
FAMILY AND PINOUT CODE (4 HEX characters) <>:
```

The Family and pinout code depends on the type of PROM programmer in use. Consult the DECprom manual and the PROM Programmer manual for more details.

Using the DECprom utility LIST command, create an unformatted memory image file (SAV type) on the host system.

Using the DECprom utility PROGRAM command, transfer the contents of the file created by the list command to the new PROM.

Example: The user wants to replace the 8 X 8K PROMs supplied with the KXJ11-CA with 8 X 32K PROMs, but still retain the native firmware functionality. The user must copy the native firmware from the 8 x 8 K PROMs into an unformatted memory image file on the host VAX/VMS system, using the DECprom LIST command.

The user removes the PROMs from their sockets on the KXJ11-CA. The user installs the low byte PROM into a compatible PROM programmer that is connected to the host VAX/VMS system.

The user then invokes the DECprom utility on the host VAX/VMS system.

```
COMMAND <HELP>: LIST
OUTPUT SPECIFICATION    <TT:>: FIRM.SAV
THE LIST FORMAT IS (AHEX/AOCTAL/ABINARY/BINARY) <AOCTAL>: BINARY
TARGET START ADDRESS    <0>: 0
TARGET STOP ADDRESS  <0>: 37776
PROM START ADDRESS (0 -> LENGTH-1) <0>: 0
ADDRESS [ 20000 : 37776 ], BIT [ 0 : 7 ], PROM COUNT IS 1
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
ADDRESS [ 20000 : 37776 ], BIT [ 8 : 15 ], PROM COUNT IS 2
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO

(*** The user must now install the high byte prom***)

MOUNT PROM AND PRESS <RETURN> TO PROCEED:
TARGET START ADDRESS (Target word boundary) <0>: EXIT
COMMAND <LIST>:
```

The native firmware has now been successfully transferred to an unformatted memory file in the host VAX/VMS system.

The user now wishes to blast the native firmware into the user-supplied 8 X 32K PROMs. To do so, the user must change parameters for the new size PROMs, and change the PROM programmer socket, etc. Then, using the DECprom PROGRAM command, the user transfers the firmware code from the file created on the host VAX to the new PROM.

```
COMMAND <HELP>: PROGRAM
INPUT FILE <>: FIRM.SAV
OLD FILE NAME (NONE) <>: NONE
FILE FORMAT (EXE/LDA/MIM/SAV/TSK/HEX/MOS) <MIM>: SAV
TARGET START ADDRESS (Target word boundary) <0>: 0
TARGET STOP ADDRESS (Target word boundary) <0>: 37776
PROM START ADDRESS (0 -> LENGTH-1) <0>: 60000
WRITE DATA TO FILE (YES/NO) <NO>: NO
ARE BLANK PROMS BEING USED (YES/NO) <YES>
ADDRESS[ 60000 : 77776 ], BIT [ 0 : 7 ], PROM COUNT IS 1
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
%PROM-S-PROGRAMMED, PROM HAS SUCCESSFULLY BEEN PROGRAMMED
ADDRESS[ 60000 : 77776 ], BIT [ 8 : 15 ], PROM COUNT IS 2
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
%PROM-S-PROGRAMMED, PROM HAS SUCCESSFULLY BEEN PROGRAMMED
COMMAND <PROGRAM>:
```

Continuing with the same example, the user now wishes to program his application into the first three quarters of the 2 X 32K PROMs. The user determines from a linker output map that the last address of the application is 136006 octal. The DECprom session begins by burning the user code:

```
COMMAND <HELP>: PROGRAM
INPUT FILE <>: MYROM.SAV
OLD FILE NAME (NONE) <>: NONE
FILE FORMAT (EXE/LDA/MIM/SAV/TSK/HEX/MOS) <MIM>: SAV
TARGET START ADDRESS (Target word boundary) <0>: 0
TARGET STOP ADDRESS (Target word boundary) <0>:137006
PROM START ADDRESS (0 -> LENGTH-1) <0>: 0
WRITE DATA TO FILE (YES/NO) <NO>: NO
ARE BLANK PROMS BEING USED (YES/NO) <NO>
ADDRESS[ 00000 : 57003 ], BIT [ 0 : 7 ], PROM COUNT IS 1
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
%PROM-S-PROGRAMMED, PROM HAS SUCCESSFULLY BEEN PROGRAMMED
ADDRESS[ 00000 : 57003 ], BIT [ 8 : 15 ], PROM COUNT IS 2
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
%PROM-S-PROGRAMMED, PROM HAS SUCCESSFULLY BEEN PROGRAMMED
COMMAND <PROGRAM>:
```

The following steps are then used to calculate and burn the checksum:

```
COMMAND <HELP>: LIST
OUTPUT SPECIFICATION <TT:>: MYROM.CC
THE LIST FORMAT IS (AHEX/AOCTAL/ABINARY/BINARY) <BINARY>: BINARY
TARGET START ADDRESS (Target word boundary) <0>: 0
TARGET STOP ADDRESS (Target word boundary) <177776>: 137774
PROM START ADDRESS (0 -> LENGTH-1) <0>: 0
ADDRESS[ 0 : 137774 ], BIT[ 0 : 7 ], PROM count is 1
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
ADDRESS[ 0 : 137774 ], BIT[ 8 : 15 ], PROM count is 2
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
COMMAND <LIST>: CALC_CHECK
INPUT FILE <>: MYROM.CC
FILE FORMAT (EXE/LDA/MIM/SAV/TSK/HEX/MOS) <SAV>: SAV
TARGET START ADDRESS (Target word boundary) <0>: 0
TARGET STOP ADDRESS (Target word boundary) <137774>: 137774
PROM START ADDRESS (0 -> LENGTH-1) <0>: 0
CHECKSUM BUFFER START LOCATION (1 --> 128) <1>: 1
COMMAND <CALC_CHECKSUM>: STORE_CHECK
CHECKSUM BUFFER START LOCATION (1 --> 128) <1>: 1
CHECKSUM COUNT (1 --> 128) <1>: 1
PROM START ADDRESS (0 -> LENGTH-1) <0>: 57777
ADDRESS[ 0 : 0 ], BIT[ 0 : 7 ], PROM count is 1
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
%PROM-S-PROGRAMMED, PROM HAS SUCCESSFULLY BEEN PROGRAMMED
ADDRESS[ 0 : 0 ], BIT[ 8 : 15 ], PROM count is 2
DO YOU WISH TO SKIP THIS PROM (YES/NO) <NO>: NO
MOUNT PROM AND PRESS <RETURN> TO PROCEED:
%PROM-S-PROGRAMMED, PROM HAS SUCCESSFULLY BEEN PROGRAMMED
COMMAND <STORE_CHECKSUM>: EXIT
```

C.2.3 PROM MAP

Figures C-1, C-2, and C-3 show how the firmware is physically mapped into the three sizes of PROMs that can be used in the KXJ11-CA PROM sites. On the left is the physical address of the PROM. On the right is the physical address(es) in the KXJ11-CA at which PROM will be visible. Multiple addresses are shown in the cases of the two smaller PROMs because of wraparound that occurs with KXJ11CA address decode.

C.2.4 CHECKSUM

The checksum for the user code is optional. If users require a checksum for their PROM code, the DECprom utility allows them to calculate it and store it. The user checksum must reside in the last user PROM location. For 8 X 16K PROMs the checksum is at offset 17777 in each PROM, and for 8 X 32K PROMs, it's at offset 57777.

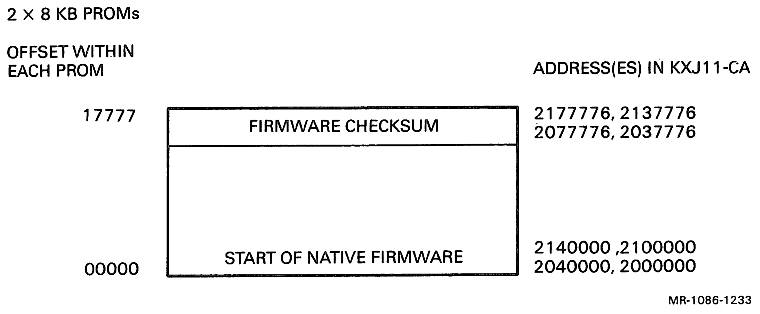


Figure C-1   2K x 8KB PROM Map

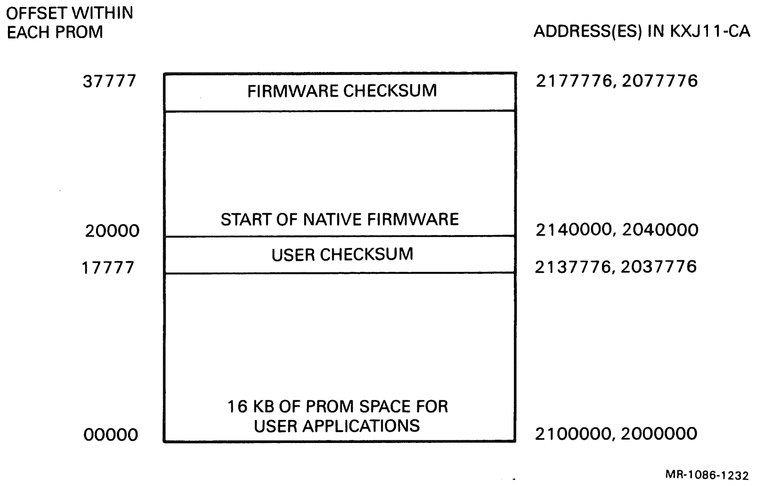


Figure C-2   2K x 16KB PROM Map

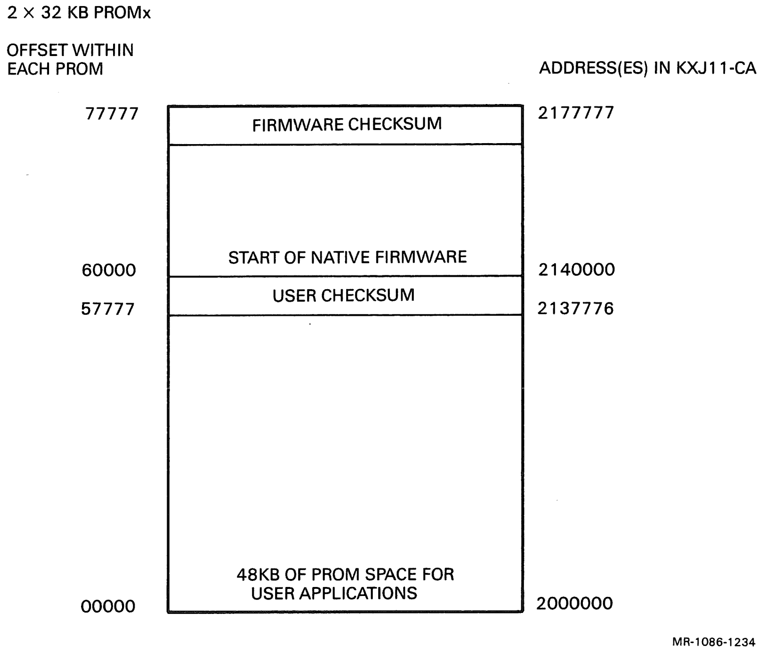


Figure C-3   2K x 32KB PROM Map

C.2.4.1 Checksum Algorithm -- The following algorithm is used to calculate the PROM checksum.

```
CHECKSUM = 0
FOR I = number of PROM addresses to be checksumed DO
        CHECKSUM = CHECKSUM + contents of address
        (high order carry from addition is discarded)
        CHECKSUM = ROTATE_LEFT_ONE_BIT
        (bit0 -> bit1, bit1 -> bit2, .... ,bit15 -> bit0 )
NEXT I
```

Or in MACRO-11:

```
     CLR    CHECKSUM
     MOV    #<ENDUSR-BGNUSR-2>/2,R0   ;NUMBER OF USER PROM WORDS
                                      ; MINUS ONE WORD FOR CHECKSUM
     MOV    #BGNUSR,R1                ;START OF USER PROM
1$:  ADD    (R1)+,CHECKSUM            ;ADD ADDR CONTENTS TO CHECKSUM
     CLC                              ;DISCARD CARRY, SET CARRY = 0
     BPL    2$                        ;BRANCH IF BIT 15 OF CS = 0
     SEC                              ;CARRY = BIT 15 = 1
2$:  ROL    CHECKSUM                  ;ROTATE CHECKSUM LEFT 1 BIT
     SOB    R0,1$                     ;LOOP CONTROL
```

Note that DECprom calculates the checksum for the PROM as a PROM set, not as individual PROMs.

APPENDIX D
BUILDING AN APPLICATION

D.1 APPLICATION BUILDING

When you build an application for the KXJ11-CA, you must specify certain switches and options. Since the application runs on the KXJ11-CA independent of the operating system environment, there is no need for a header, stack area, or memory management. Therefore, when building an application, the negated header switch (/-HD) is attached to the image file specification. This switch suppresses the header within the image. To suppress memory management, the negated memory management switch (/-MM) is attached to the image file specification. This switch specifies that the application will run in a system that does not have memory management hardware.

To suppress the stack area in the task builder command sequence during option input, STACK=0 should be used. If you need to specify a stack with your application, you should do so within the application itself. Since the KXJ11-CA application starts up as unmapped, the application is initially bound to physical memory. The application must be installed at the same memory address for which it was built. Therefore, during option input, use the PAR ption to specify the base address and length of an application.

The following examples build a 24 kilobyte application code file named APPCODE.

MCR Example:

```
MCR> TKB<RET>
TKB> APPCODE/-HD/-MM=APPCODE<RET>
TKB> /<RET>
Enter Options:
TKB> STACK=0<RET>
TKB> PAR=DUMCODE:0:60000<RET>
TKB> //<RET>
```

DCL Example:

```
DCL> LINK APPCODE/NOHEADER/NOMEMORY_MANAGEMENT/OPTIONS<RET>
Options? STACK=0<RET>
Options? PAR=DUMCODE:0:60000<RET>
```

INDEX

Digital Equipment Corporation • Marlboro, MA 01752