KXT11-CA Single-Board Computer User's Guide

digital

EK-KXTCA-UG-002

KXT11-CA
Single-Board Computer
User's Guide

digital™

EK-KXTCA-UG-002

KXT11-CA Single-Board Computer User's Guide

Prepared by Educational Services
of
Digital Equipment Corporation

1st Edition, May, 1984
2nd Edition, September 1984

The material in this manual is for informational purposes and is subject to change without notice.

Digital Equipment Corporation assumes no responsibility for any errors that may appear in this manual.

The manuscript for this book was created on a DIGITAL Word Processing System and, via a translation program, was automatically typeset on DIGITAL's DECset Integrated Publishing System. Book production was done by Educational Services Development and Publishing in Marlboro, MA.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

CONTENTS (Cont)

Page

CONTENTS (Cont)

FIGURES

FIGURES (Cont)

FIGURES (Cont)

FIGURES (Cont)

FIGURES (Cont)

Figure No. Page

TABLES

Table No. Title Page

TABLES (Cont)

PREFACE

This user's guide is intended for hardware, software, and field engineers as an aid in the application/use, programming, and maintenance of the KXT11-CA single board computer (SBC). KXT11-CA SBC users should be familiar with Digital's Q-Bus family of products, the PDP-11 instruction set, general microprocessor operations, and microcomputer system applications.

GUIDE STRUCTURE

Chapter 1 provides a brief description of the KXT11-CA SBC, a list of features, and a simplified block diagram with description.

Chapter 2 supplies an overview of the two operating modes and discusses the hardware and firmware associated with each. Included is the 22-bit address/16-bit data Q-Bus interface block diagram.

Chapter 3 provides detailed functional descriptions of the KXT11-CA SBC operations with timing diagrams and a detailed functional block diagram.

Chapter 4 details the macro-ODT, bootstrap, and selftest routines performed by the native firmware of the KXT11-CA SBC and how each routine is implemented.

Chapter 5 describes all the software programmable registers located on the KXT11-CA SBC and the register contents after initialization.

Chapter 6 describes how to configure, install, and maintain the KXT11-CA SBC. Details include operational features, external cable connections, and maintenance verification procedures.

Appendix A lists the PDP-11 instruction set and general addressing modes.

Appendix B is a list comparing KXT11-CA SBC operations with Digital's other Q-Bus family members.

Appendix C lists the performance parameters of the on-board T-11 microprocessor (CPU) for the entire instruction set.

Appendix D includes a complete set of KXT11-CA SBC schematic drawings.

Appendix E provides a listing of the KXT11-CA address environment.

Appendix F provides details on constructing the four I/O loopback connectors.

Appendix G is a glossary of terms used in this guide.

The Index is an alphabetical listing of the guide's content.

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The KXT11-CA (M8377) single-board computer (SBC) is shown in Figure 1-1. It is a quad-height, extended-length, single-width board that executes the PDP-11 instruction set (see Appendix A). The KXT11-CA SBC operates as a peripheral processor, or a standalone single-board computer. In addition, the KXT11-CA SBC supports the complete Q-Bus extended bus interface as a bus slave and a DMA master. This enables the KXT11-CA SBC to communicate with most of Digital's large family of modules (see Chapter 3) described in the *Microcomputer Interfaces Handbook* and the *Microcomputers and Memories Handbook.*

1.2 KXT11-CA HARDWARE

The KXT11-CA simplified functional block diagram (Figure 1-2) provides an overview of the board's functions and how they are related. The KXT11-CA SBC has a 16-bit microprocessor interconnected to the following.

- Three serial I/O ports
- One 20-line parallel I/O port
- A two-channel 16-bit DMA transfer controller (DTC).
- Three programmable timers
- Local status/control registers
- A memory consisting of RAM, user socket sites, and native PROM firmware via the local 16-bit address/data bus lines

The internal bus operates as either a 16-bit or an 8-bit bus.

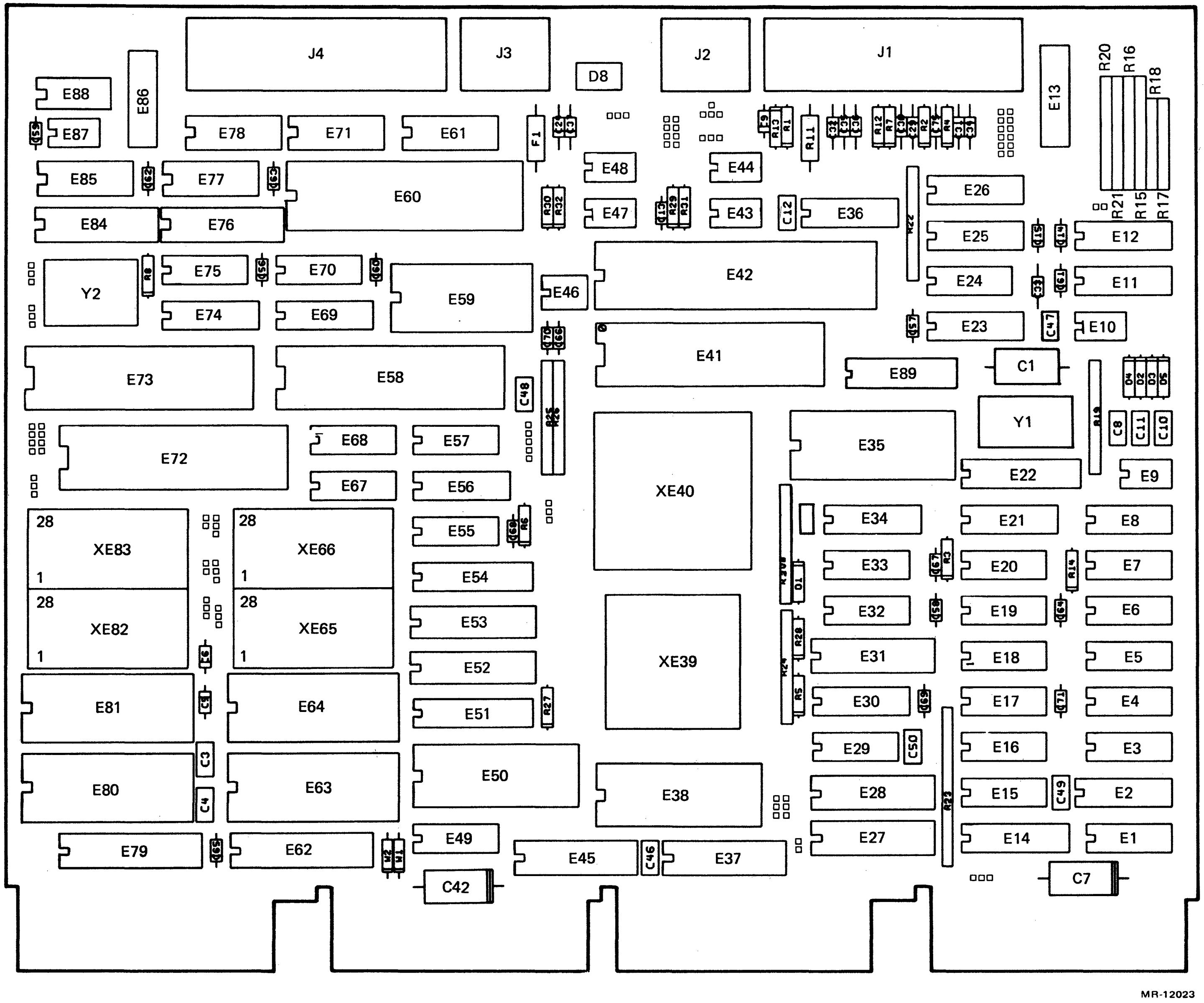


Figure 1-1   KXT11-CA SBC Layout

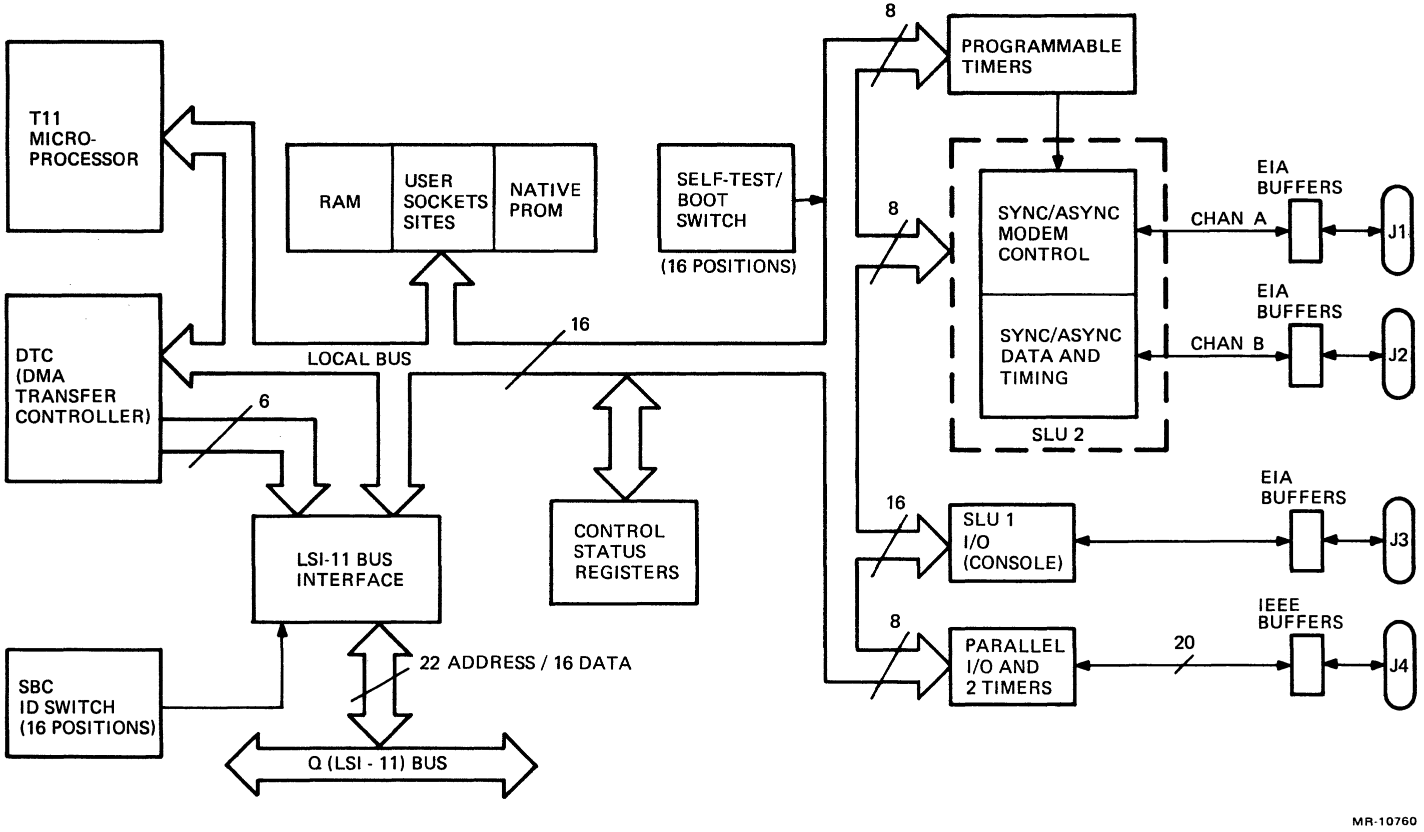


Figure 1-2 KXT11-CA SBC Simplified Block Diagram

1.3 KXT11-CA FEATURES

The features of the KXT11-CA SBC include the following.

- T-11 16-bit microprocessor emulating the PDP-11 instruction set
  - Direct addressing of 64 K bytes
- Local memory
  - On-board 32 Kb of static read/write memory (RAM)
  - Sockets for up to 32 Kb of additional PROM or static RAM.
  - 8 Kb of native firmware
- Q-Bus extended bus interface
  - A 16-word, two-port register file (TPR)
  - A local-to-Q-Bus interrupt mechanism.
- A direct transfer controller (DTC) for memory or I/O data transfers
  - Local-to-local
  - Local-to-Q-Bus
  - Q-Bus-to-local
  - Q-Bus-to-Q-Bus
- Four local registers for status and control
  - Four programmable LED indicators for diagnostic fault isolation
  - Selftest mode switch
- One asynchronous serial (console) I/O line unit
  - EIA RS-422/RS-423/RS-232 compatible
  - Programmable baud rates of 300 to 38,400 bits/baud
- One synchronous/asynchronous serial line unit
  - Full modem support
  - RS-449 (CCITT V.24) type SR (send-receive)
  - EIA RS-422/EIA RS-232C compatible
  - Programmable baud rates of 110 to 56,000 bits/baud
- One asynchronous secondary channel
  - RS-449 (CCITT V.24) type DT (data and timing only)
  - EIA RS-423/RS-422 compatible
  - Asynchronous 32-node party line electrical interface
  - Programmable baud rates of 110 to 56,000 bits/baud
- Three programmable timers/counters for serial communications
- One parallel I/O interface
  - Two bidirectional 8-bit input/output ports
  - One 4-bit control port
  - Three 16-bit programmable interval timers/counters
  - IEEE 488 Electrical Interface compliance

1.4 KXT11-CA OPERATING MODES

The KXT11-CA SBC can be operated in either of two modes: standalone or peripheral processor. A general description of both operating modes is included in the following sections.

1.4.1 Standalone Mode

In the standalone mode, the KXT11-CA SBC can be used as a dedicated controller or general-purpose microcomputer. When physically connected to the Q-Bus via its backplane, the KXT11-CA SBC's local address/data bus is isolated from the Q address/data bus. However, the power supply and ground signals can be supplied by the Q-Bus. If the KXT11-CA SBC is not connected to a Q-Bus, power and ground signals must be externally supplied.

All communications between local and global addresses are done via one of the four communication I/O ports.

1.4.2 Peripheral Processor Mode

As in the general-purpose, single-board computer operation, the microprocessor, DTC, and I/O port operations are identical. In addition, the KXT11-CA SBC is capable of interfacing with Digital's Q-Bus (22-bit address) via backplane connections. The Q-Bus interface allows the existing system to off-load some work onto a maximum of 14 KXT11-CA SBCs.

Q-Bus interfacing is provided by KXT11-CA hardware, firmware, and software. The Q-Bus interface allows the internal SBC's multiplexed address/data bus access, via the external Q-Bus 22-bit memory space, to the direct transfer controller (DTC) interface for moving data parameters. The bus master (arbiter) uses the Q-Bus to transmit commands, receive status, and transfer blocks of control data to the 16-word, two-port register file (TPR). In addition, the KXT11-CA SBC includes Q-Bus interrupt hardware for signaling the Q-Bus arbiter. The interrupt is initiated when service by the existing system is required.

In the peripheral processing mode, DTC operations are not limited to transferring data between local 16-bit addresses. Transferring of data also occurs between the following.

- A local 16-bit address and a 22-bit Q-Bus address
- A 22-bit Q-Bus LSI address and a 16-bit local address
- Two 22-bit Q-Bus addresses

The DTC transfers data using flowthrough operations allowing up to 64 Kb to be transferred during one DMA operation.

1.5 KXT11-CA SPECIFICATIONS

Physical

Height (quad)	26.6 cm (10.5 in)
Length (dual)	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

Power Supply

+5 V ± 5%	3.5 A (typical), 4 A (maximum)
+12 V ± 5%	60 mA (typical) used by on-board circuitry; 2 A (maximum) includes current provided to outside interface through pin 10 of the serial I/O connector for operating the DLV11-KA EIA-20 mA converter option.

NOTE

The +12 V typical current is measured with no connections at pin 10 of the serial I/O connector (fused line).

Battery Backup

+5 V ± 5%	12 mA (typical), 20 mA (maximum)

Bus Loading

AC loads	2.7
DC loads	1.0
PIO	
EIA	

Environmental

Temperature

Storage	–40°C to 66°C (–40°F to 150°F)
Operating	5°C to 60°C (41°F to 140°F)

NOTE

The module must be brought into the operating temperature environment and allowed to stabilize for five minutes before operating.

Relative Humidity

Storage	10% to 95% (no condensation)
Operating	10% to 95% (no condensation)

Altitude

Storage	Up to 15 km (50,000 ft)
Operating	Up to 15 km (50,000 ft) (90 mm mercury minimum)

NOTE

Lower the maximum operating temperature by 1°C (1.8°F) for each 300 m (1,000 ft) of altitude above 2.4 km (8,000 ft).

Air quality
Air must be noncaustic.

Airflow (operating)
There must be enough airflow to limit the input to output temperature rise across the module to 5°C (9°F) when the input temperature is 60°C (140°F). For operation below 55°C (131°F), there must be enough airflow to limit the input to output temperature rise across the module to 10°C (18°F) maximum.

1.6 RELATED DOCUMENTS

This user's guide is the primary reference document for the KXT11-CA SBC. Important information about other Q-Bus-compatible products may be found in the publications listed in Table 1-1.

Table 1-1 Related Documentation

Title	Document Number
Microcomputers and Memories Handbook	EB-20912-20
Microcomputer Interfaces Handbook	EB-23144-18
PDP-11 Architecture Handbook	EB-23657-18
Chipkit Users Manual	EJ-17475
μ/T11 User's Guide	EK-DCT11-UG
TU58 Technical Manual	EK-OTU58-TM
KXT11-CA ROM Listing	EK-KXTCA-HR

These documents can be ordered from:

Digital Equipment Corporation
Accessories and Supplies Group
Cotton Road
Nashua, NH 03060

Attention: Documentation Products
Telephone: 1-800-258-1710

Additional Documents

For backup information involving chip-level detail, the following publications can be ordered.

Z8036 Counter/Timer and Parallel I/O Unit, 1982 Edition
Z8016 DMA Controller, 1982 Edition

Zilog
1315 Dell Ave.
Campbell, CA 95008

Attention: Corporate Publications

μPD7201 Technical Manual, 1981 Edition

NEC Electronics, Inc.
One Natick Executive Park
Natick, MA 01760

Attention: Corporate Publications

CHAPTER 2
KXT11-CA SBC ARCHITECTURE

2.1 INTRODUCTION

The KXT11-CA SBC architecture enables the KXT11-CA SBC to be configured in either of two basic modes: standalone or peripheral processor. An ID switch, mounted on the SBC's edge, enables operating mode selection according to the user's application. This chapter describes how each mode is used and what portion of the SBC's native firmware is implemented. In addition, the protocols invoked by the native firmware for each mode are discussed.

2.2 STANDALONE MODE

The KXT11-CA SBC contains all the necessary hardware and native firmware needed to operate as a complete standalone single-board microcomputer. In the standalone mode the KXT11-CA SBC can be used for dedicated computer tasks, operating completely independent of any Q-Bus signals.

2.3 STANDALONE HARDWARE

The KXT11-CA SBC hardware includes the following.

- A CPU for instruction manipulations
- RAM for temporary data storage
- ROM for execution of on-board selftest/boot/ODT (octal debugging technique) routines
- User socket sites for executing application code
- Direct transfer controller (DTC) for executing DMA transfers
- Control and status registers for system operations
- ID switch for selecting operating mode
- Selftest/boot switch for selecting selftest/boot routines
- Four I/O ports for communications with a variety of I/O devices

See Figure 2-1 for the simplified KCT11-CA SBC hardware block diagram.

2.3.1 Backplane Connections

The KXT11-CA SBC can be operated in the standalone mode while physically connected, via the backplane connections, to the Q-Bus. In the standalone mode the KXT11-CA SBC bus transceivers are disabled, inhibiting any local references from the Q-Bus to be detected and local references to the Q-Bus to be aborted. Since references between local and Q-Bus spaces are electrically isolated, the KXT11-CA SBC can operate completely independent of any Q-Bus signals.

2.3.2 Power Connections

In the standalone mode, when the KXT11-CA SBC is physically connected to an Q-Bus, the +5 Vdc, +12 Vdc and system ground can be received either from the Q-Bus or from an external source. If the KXT11-CA SBC is not physically connected to the Q-Bus, the +5 Vdc, +12 Vdc, and system ground must be externally supplied. In addition, the on-board RAM (if installed in user sockets) can be backed up by battery by supplying a minimum of +2 Vdc either externally or via the Q-Bus. See Paragraph 1.5 for the battery back-up voltage specifications.

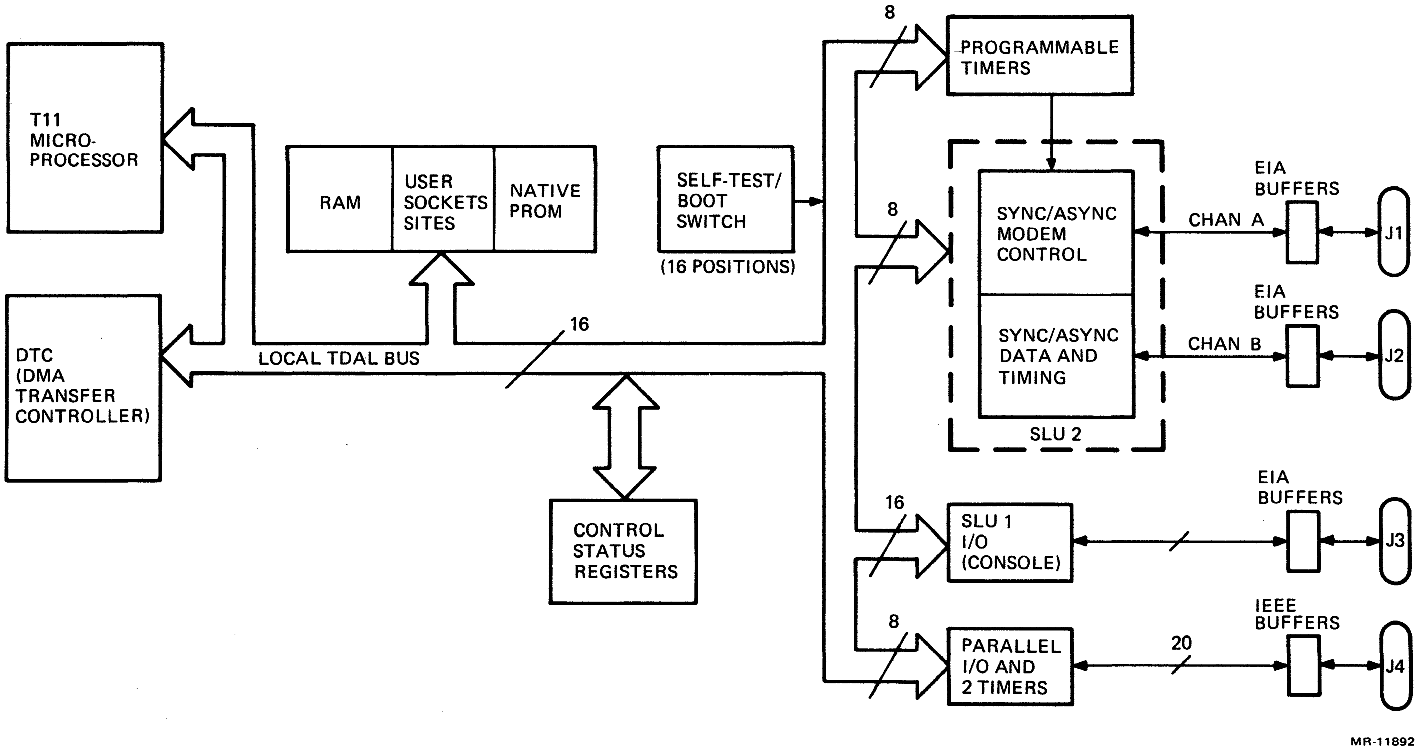


Figure 2-1 Simplified KXT11-CA SBC Block Diagram, Standalone Mode

2.4 DMA OPERATIONS

The direct transfer controller (DTC) is a two-channel controller capable of transferring large blocks of data (up to 64 Kb), using a DMA operation, with minimal microprocessor involvement. All DMA transfers are initiated locally by the T-11 microprocessor. DMA transfers can occur between 16-bit local addresses and support hardware requests from the two-channel synchronous/asynchronous I/O port controller or the parallel I/O port controller. DMA operations can be interleaved between the DTC's two channels, between one channel and the local processor, or occur in various bursts.

These transfers are performed in a two-step sequence called flowthrough transactions, where a source is read and a destination is written. Flowthrough transactions can occur between all local memory combinations. The data is temporarily held in DTC registers between the source and destination portions of the transfers. On the next local bus cycle, the DTC reads the source and temporarily stores the data. During the following local bus cycle, the destination is written and the data is transferred to the destination.

2.5 NATIVE FIRMWARE

The KXT11-CA SBC contains all the necessary native firmware for controlling on-board basic operations. These include selftest/boot/ODT codes for implementing the start-up and communications protocols. All KXT11-CA SBC registers are accessed using a 16-bit even boundary octal address. Data information can be in either 16-bit word or 8-bit byte format. Refer to Figure 2-2 for the KXT11-CA native firmware chart.

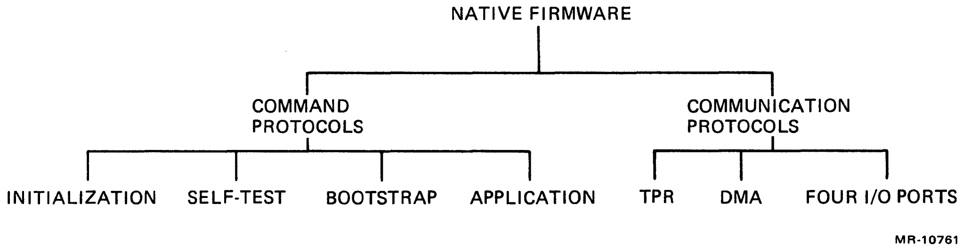


Figure 2-2 KXT11-CA SBC Native Firmware

2.5.1 Selftest/Boot Routines

Selftest/boot routines are selected via the hardware selftest/boot switch. The native firmware provides three selectable functions as follows.

1. The user can select one of three selftest functions. One function disables the selftests; the second performs a normal selftest; and the third performs selftests repeatedly.
2. If the user's application program is stored in an off-board I/O device, any one of the four I/O ports can be selected as the communication path for booting the application program into the local RAM space.
3. The user's application program can be booted from two possible memory sources. One memory source is located in a section of the user's PROM, and the other memory source is located off-board in an I/O device.

2.5.2 Start-Up Protocols

The start-up protocols are a subset of the ODT command protocols. Start-up protocols cover three separate SBC operations: initialization, selftest and bootstrap. Each step in these operations is followed to ensure proper operating procedures. The start-up protocols (Figure 2-3) are software implemented through the native firmware.

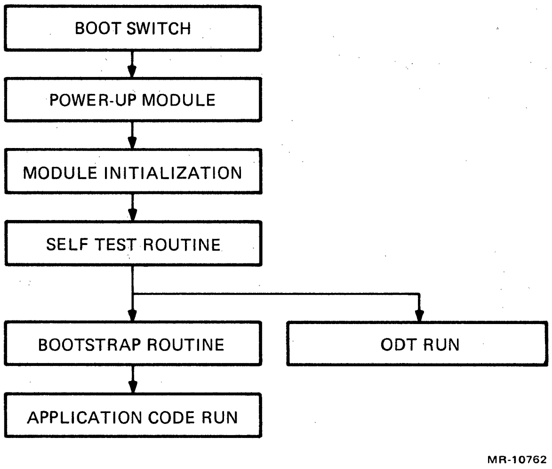


Figure 2-3 Start-Up Protocol Flow Diagram

2.5.2.1 Initialization – Before power is applied to the KXT11-CA SBC, the selftest/boot routines are selected via the selftest/boot switch. During power-up, the T-11 initializes all on-board chips according to the native firmware ROM. This includes clearing some registers and writing into others.

2.5.2.2 Selftest – After the KXT11-CA SBC is initialized, the native firmware performs the selected selftest routines. The selftest routine invoked is dependent on the selftest/boot switch position. These routines verify proper operations of many on-board SBC functions. Selftest routines can be performed as a portion of the power-up procedure or can be implemented by native firmware directives.

Selftest routines check for two types of errors: nonfatal and fatal. For nonfatal errors the boot routine continues and the LEDs display the error encountered , after booting SBC operations are halted. When a fatal error is detected, the boot routine is immediately stopped with the LEDs displaying the error encountered. Refer to Chapter 6 for detailed LED error information.

2.5.2.3 Boot – If the selftest routine is passed, a boot routine is initiated by the firmware according to the selftest/boot switch selection. The application software code is then booted into local RAM space. The boot routine is done via a preselected I/O port. Once the boot routine is complete, control of KXT11-CA SBC is transferred to application software. The native firmware supports system level exception handling while the application program is running.

2.5.3 Communication Protocols

In the standalone mode communication protocols exist for all four I/O ports. Two protocols are used: one during boot routines and the other during the running of the application program. Three serial I/O ports are available for boot routines and all four I/O ports are available for communications during the running of application programs.

2.5.3.1 I/O Port Booting – The KXT11-CA SBC firmware supports communications for primary bootstrap routines to load the application program into local RAM space. Any one of the three serial I/O ports can be used for this purpose – the asynchronous SLU1 (serial line unit) or either of the two SLU2 channels (A and B).

SLU1 can be used to load the application program from the off-board TU58 I/O device. The TU58 is a mass storage cassette tape device that can be used to store and down-line-load user application programs. To ensure proper operation between the TU58 and the KXT11-CA SBC, both baud rates must match. See the *TU58 Technical Manual* for baud rate details. Also, refer to Paragraph 6.3.11 in this user guide for SLU1 baud rate details.

SLU2 can be used to boot the application program in either asynchronous or synchronous mode. The controller for SLU2 consists of a two-channel device, channels 1 and 2. Both channels are separate I/O ports and can be used for either asynchronous or synchronous communications. The necessary firmware for using SLU1 or 2 as communication I/O ports during boot routines is supported by the local ROM and is implemented through the selftest/boot select switch. Figure 2-4 shows the I/O port boot path.

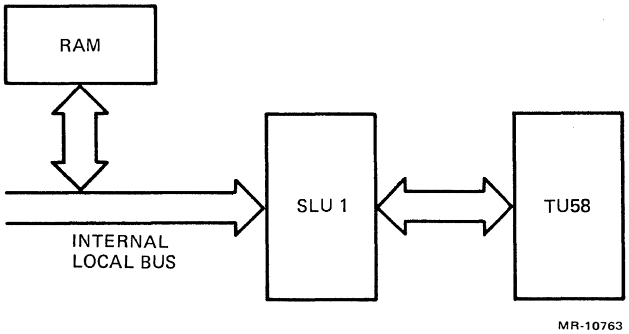


Figure 2-4 I/O Port Boot Path

2.5.3.2 I/O Port Communications – After the application program is booted into local RAM space, communications via one of four I/O ports is supported and controlled by the application program. The user's application program must contain the necessary code to implement communications via the selected I/O port. Since the T-11 microprocessor is interrupt driven, the KXT11-CA SBC can support simultaneous communications via all four I/O ports. This means that each time data is received by an I/O port, an interrupt to the T-11 is generated. Servicing of these bus interrupts is on a priority basis.

2.6 PERIPHERAL PROCESSOR MODE

The KXT11-CA SBC uses the same hardware circuits and performs all of the operations as in the standalone mode. Some additional hardware circuitry is used to interface the KXT11-CA SBC to the Q-Bus extended bus. This hardware enables the KXT11-CA SBC to operate as a peripheral processor or an intelligent processor on Digital's Q-Bus. Digital's Q-Bus extended bus consists of a multiplexed 22-bit address/16-bit data lines. The main purpose of this mode is to allow the system currently existing on the Q-Bus to off-load some work onto the KXT11-CA SBC.

2.7 PERIPHERAL PROCESSOR ARBITER

When the peripheral processor mode is selected, the KXT11-CA SBC operates as a bus slave device requiring the operation of an arbiter (master) device on the bus. A bus arbiter is a CPU having the ability to handle bus initialization, arbitrate DMA operations, and service Q-Bus interrupts. A bus master can be any device on the bus having control over current bus operations. A bus arbiter is capable of being both arbiter and bus master. A KXT11-CA SBC can be a bus master during DMA operations but can never be bus arbiter. Any and all transactions between the arbiter and the IOP must be initiated by the bus arbiter. The SBC must always receive a command before any data is made available to the Q-Bus or any data is taken from the Q-Bus arbiter.

2.8 PERIPHERAL PROCESSOR/IOP HARDWARE

KXT11-CA SBC hardware interface includes 22-bit bus transceivers, a 16-bit by 16-word two-port RAM (TPR) file, and a Q-Bus interrupt register (QIR). The bus transceivers are the primary interface for transferring 22/16-bit multiplexed address/data lines across the Q-Bus. A unique two-port RAM file supports system control, status, and data transactions between the Q-Bus and the on-board T-11 microprocessor. The DTC transfers large blocks of data using the local 22-bit bidirectional transceivers for DMA operations. The Q-Bus interrupt register (QIR) allows the SBC to send 8-bit level 4 interrupt vectors to the bus arbiter. Figure 2-5 is a block diagram showing the peripheral processor interface.

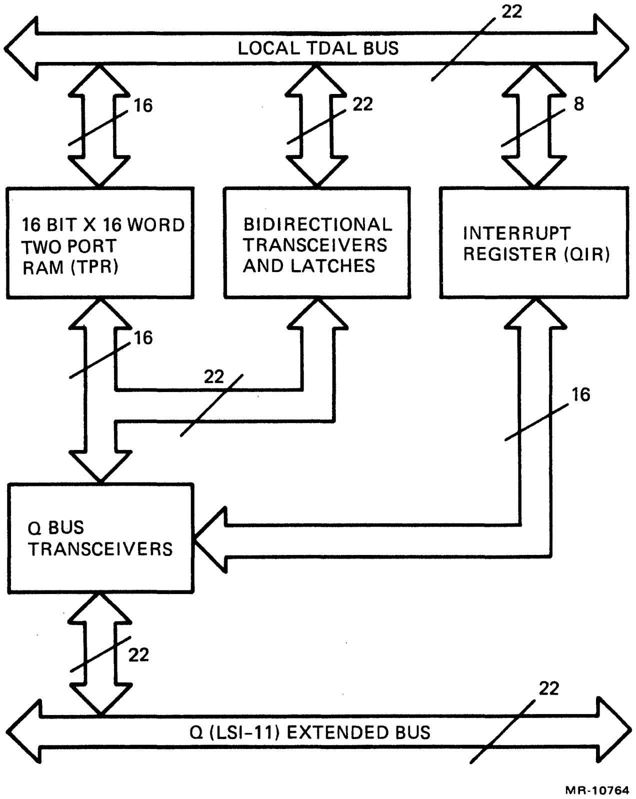


Figure 2-5 Peripheral Processor/IOP Interface

2.8.1 Backplane Connections

Backplane connections provide the signal path between the KXT11-CA hardware and the Q-Bus. This signal path carries bidirectional, multiplexed, 22-bit address/16-bit data and control signals.

2.8.2 Power Connections

The Q-Bus supplies power (+5 Vdc, +12 Vdc, and system ground) to the KXT11-CA SBC. In addition, the battery backup voltage for the on-board RAM can be supplied through the Q-Bus. This battery backup voltage ensures RAM data retention when power is removed from the module.

2.9 NATIVE FIRMWARE

In addition to using standalone mode native firmware, the peripheral processor mode uses some additional native firmware for two-port RAM operations, Q-Bus arbiter interrupts, and SBC interrupts.

The native firmware ROM contains the necessary selftest/boot/ODT codes for implementing, selftesting, SBC initializing, application program booting, and application program running.

2.9.1 Two-Port RAM (TPR)

All commands between the arbiter and the KXT11-CA SBC are passed through the SBC's two-port RAM file. The TPR is a 16-bit by 16-word register file supporting simultaneous read references and concurrent read/write combinations from both local and Q-Bus memory spaces.

The first four registers in the file are used exclusively for local SBC system control. Support for the system control registers consists of supplying local status information to the bus arbiter. The remaining 12 registers can be used by the SBC's application software and the system applications software.

2.9.2 DMA OPERATIONS

The direct transfer controller (DTC) is the device used to transfer large blocks of data, using DMA operations, with minimal local processor involvement. A DMA operation can be initiated locally by the processor or by the Q-Bus arbiter.

When initiated by the bus arbiter, the local processor receives command words and DMA transfer parameters via the TPR file. After receiving transfer commands and parameters, the local processor initiates the proper DTC operations. DMA operations occur between a 16-bit local address to a 22-bit global address, a 22-bit global address to a 16-bit local address, a 22-bit global address to a 22-bit global address, or a 16-bit local address to a 16-bit local address.

2.9.3 Arbiter To SBC Interrupts

During or after initialization, the Q-Bus arbiter can interrupt SBC operations by writing a command into the TPR's first register. This is the only register available to the bus arbiter prior to complete initialization. Every time the bus arbiter writes into this register, an interrupt is generated causing a nonmaskable trap to the restart address of the T-11 chip. Native firmware then interprets the source of the interrupt and implements the proper action. This register is intended to be the priority command channel for the arbiter. Figure 2-6 shows the arbiter to KXT11-CA interrupt cycle.

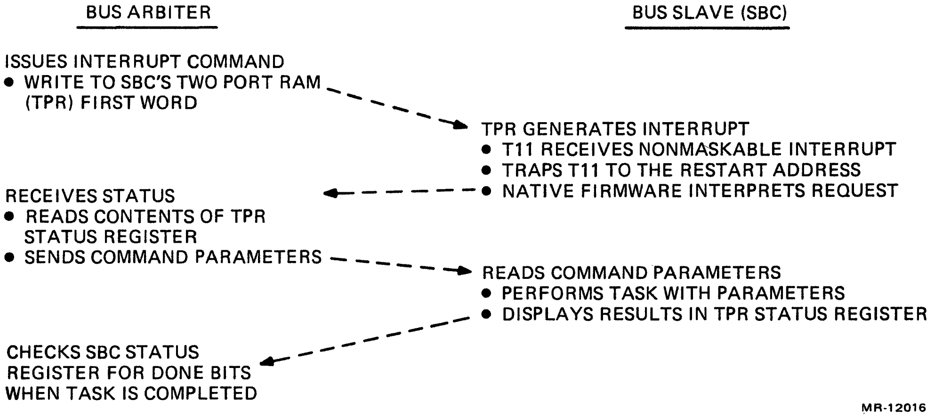

BUS ARBITER
BUS SLAVE (SBC)
ISSUES INTERRUPT COMMAND
• WRITE TO SBC'S TWO PORT RAM (TPR) FIRST WORD
TPR GENERATES INTERRUPT
• T11 RECEIVES NONMASKABLE INTERRUPT
• TRAPS T11 TO THE RESTART ADDRESS
• NATIVE FIRMWARE INTERPRETS REQUEST
RECEIVES STATUS
• READS CONTENTS OF TPR STATUS REGISTER
• SENDS COMMAND PARAMETERS
READS COMMAND PARAMETERS
• PERFORMS TASK WITH PARAMETERS
• DISPLAYS RESULTS IN TPR STATUS REGISTER
CHECKS SBC STATUS REGISTER FOR DONE BITS WHEN TASK IS COMPLETED
MR-12016

Figure 2-6 Bus Arbiter to KXT11-CA SBC Interrupt Cycle

After complete SBC initialization, interrupts can be sent from the arbiter into either of two register locations in the two-port RAM file. The firmware associated with these two registers interpret the interrupt as a level 5 interrupt. After receiving this type of interrupt the processor reads the contents of the first TPR register for the command word. These two registers are intended to be secondary command channels for the arbiter.

2.9.4 KXT11-CA SBC To Arbiter Interrupts

The KXT11-CA SBC must signal the bus arbiter every time it requires attention or completes a task. Native firmware supports this signal by providing a word addressable write-only Q-Bus interrupt register (QIR). The local processor can write vector values between 0 and 1774 (octal) into this register and the QIR asserts Q-Bus interrupt request line. The bus arbiter responds by requesting the interrupt register contents. After receipt of the interrupt acknowledge, a control signal from the bus arbiter, the contents of the QIR is driven onto the bus. Upon receipt of the vector, the arbiter can service the SBC. Refer to Figure 2-7 for the KXT11-CA to arbiter interrupt cycle.

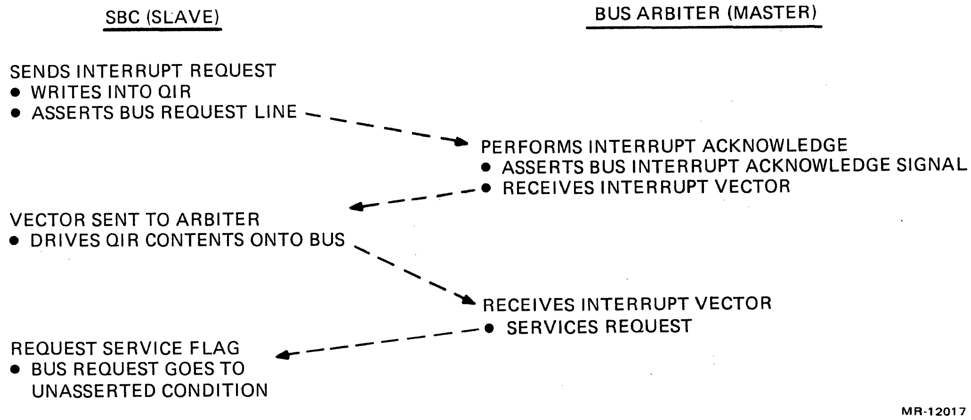

SBC (SLAVE)
BUS ARBITER (MASTER)
SENDS INTERRUPT REQUEST
• WRITES INTO QIR
• ASSERTS BUS REQUEST LINE
PERFORMS INTERRUPT ACKNOWLEDGE
• ASSERTS BUS INTERRUPT ACKNOWLEDGE SIGNAL
• RECEIVES INTERRUPT VECTOR
VECTOR SENT TO ARBITER
• DRIVES QIR CONTENTS ONTO BUS
RECEIVES INTERRUPT VECTOR
• SERVICES REQUEST
REQUEST SERVICE FLAG
• BUS REQUEST GOES TO UNASSERTED CONDITION
MR-12017

Figure 2-7 KXT11-CA SBC to Bus Arbiter Interrupt Cycle

2.9.5 Selftest/Boot Switch

Selftest/boot routines are selected via the hardware selftest/boot switch. The native firmware supports these routines by providing two selectable functions as follows.

1. The user can select one of three selftest functions. One function does not perform a selftest; the second performs a selftest; and the third performs extended selftests.

2. The user's application program can be booted from two possible memory sources. One memory source is located in the user's PROM; the other is located off-board in an I/O device.

2.9.6 Command Protocols

When operating in the peripheral processor mode, portions of SBC native firmware are used for implementing command and communications protocols. These protocols cover all basic SBC operations. Included under command protocols are the initialization, selftest, bootstrap, and application program routines. Communication protocols are implemented using one of three separate I/O ports.

SBC initialization, selftest, and boot routines can be entirely initiated and directed by the Q-Bus arbiter. This includes both the selftest and bootstrap operations. Boot operations provide the loading of an application program from a Q-Bus memory space or local memory space to local SBC RAM space.

2.9.6.1 Start-Up – Prior to applying SBC power, both SBC ID and the selftest/boot switches are set. The SBC ID switch selects the peripheral mode providing proper I/O page addressing. The selftest/boot switch selects the routines to be implemented. Applying power to the bus supplies the Q-Bus with +5 Vdc, +12 Vdc and system ground. During power-up the T-11 microprocessor initializes all on-board chips according to the native firmware ROM. This includes the clearing of some registers and writing into others.

2.9.6.2 Selftest – After SBC initialization is complete, the native firmware invokes the selected selftest. This test verifies the operations of as many specific basic SBC functions as possible. The test performed is dependent on the selftest/boot switch position. One selftest function initiates booting from the Q-Bus thereby disabling any selftests. Another selftest function performs the test as a portion of the power-up procedure or is implemented by directives from the native firmware. The last selectable selftest function does not perform a selftest but waits for a command from the Q-Bus arbiter.

Selftest routines check for two types of errors: nonfatal and fatal. For nonfatal errors the boot routine continues with the LEDs displaying the last error encountered. After booting is complete a nonfatal error will halt any further SBC operations. When a fatal error is detected, the boot routine is immediately stopped with the LEDs displaying the last fatal error encountered. Refer to Chapter 6 for detailed LED error information.

2.9.6.3 Boot – In addition to the standalone mode booting procedure described in Paragraph 2.5.1.2, booting from a Q-Bus memory space is supported. One boot routine does not perform any booting but places the SBC in a hold state, waiting for a command from the Q-Bus arbiter. Another boot routine initiates the selftests and places the SBC in a hold state waiting for a command, via the TPR's first word, from the Q-Bus arbiter. When the command is received, booting of the application program is done via a DMA operation. The application program is then transferred into the local RAM space. Once the application program is booted, application software controls SBC operations.

2.9.7 Communication Protocols

Three data paths are supported by the peripheral processor mode. One path uses the TPR register file and the second data path bypasses the TPR using local bidirectional latched transceivers. The third path occurs between the SBC and off-board I/O devices using one of the four I/O ports.

2.9.7.1 Two-Port RAM (TPR) – Six words in the two-port RAM form the main communication path for transferring data between the SBC and the Q-Bus. This path is used when messages are small in length. Normally, these file words are used in conjunction with the four system control words. After sending a command word, six file words are used to pass associated parameters. Considerable microprocessor time is required for this type of data transfer. When large blocks of data are transferred, a DMA operation is used, requiring minimal microprocessor time.

2.9.7.2 DTC Transfers – For transferring large blocks of data during DMA operations, a communication path requiring minimum microprocessor time is provided. The transfer of large blocks of data is initiated locally (T-11) or globally (arbiter), both using the DTC for performing these transfers. Locally, all DMA transfer commands and parameters are sent directly to the DTC from the T-11. Global transfer commands and parameters are first sent to the TPR's first register. Then, the transfer commands and parameters are received by the T-11 and passed to the local DTC (DMA controller). Finally, the DTC initiates the DMA operation. Data being transferred is latched, via DTC control, onto the local bus via internal bus transceivers. This path uses the Q-Bus transceivers bypassing the TPR.

2.9.7.3 I/O Port Booting – The KXT11-CA SBC supports all the same booting protocols and procedures as in the standalone mode. (See Paragraph 2.5.3.1.) In addition, the peripheral processor mode supports boot routines initiated and directed from the Q-Bus arbiter. To do this, the bus arbiter writes into the first register of the local two-port RAM. Associated parameters are sent via the TPR's secondary control registers.

2.9.7.4 I/O Port Communication – After the application program is booted into the local RAM space, communications via any I/O port is supported and controlled by the native firmware through the application software. The user's application software must contain the necessary commands to implement communications via selected I/O ports. Since the T-11 microprocessor is interrupt driven, the KXT11-CA SBC can simultaneously support communications over all four I/O ports. This means that during each interval data is received by an I/O port, an interrupt to the T-11 is generated.

CHAPTER 3
THEORY OF OPERATION

3.1 INTRODUCTION

This chapter provides a detailed explanation of KXT11-CA hardware operation from the perspective of the logic designer. It is useful for an understanding of the SBC to the chip level.

NOTE

The negated or inverse signal is designated by letter after the signal name. For example, PI H is normally low and asserted high when activated; RAS L is normally high and asserted low when activated. This convention is used throughout this chapter.

The KXT11-CA SBC functional block diagram (Figure 3-1, Sheets 1, 2, and 3) provides an overview of the SBC functions and how they are related. The main components of the SBC are shown on Sheets 1 and 2. The SBC has a microprocessor interconnected to the serial line units, RAM memory, ROM memory, direct transfer controller (DTC), and parallel I/O interface via internal TDAL and IIDAL buses. The address bus, memory address decode function, Q-Bus bus interrupt register, two-port RAM with logic sequencer, and the interrupt control function are also shown on Sheets 1 and 2.

The microprocessor support functions and the Q-Bus interface control functions are detailed on Sheet 3 of Figure 3-1. Also included are the power-up, clock control, ready, DMA, and bus interrupt function. The DMA, interrupt control, read/write and other functions are used to interface the Q-Bus to the microprocessor.

The functional descriptions used in this chapter define the microprocessor and the input/output signals associated with its operation. The support functions, Q-Bus interface functions, and the remaining devices are also described in detail.

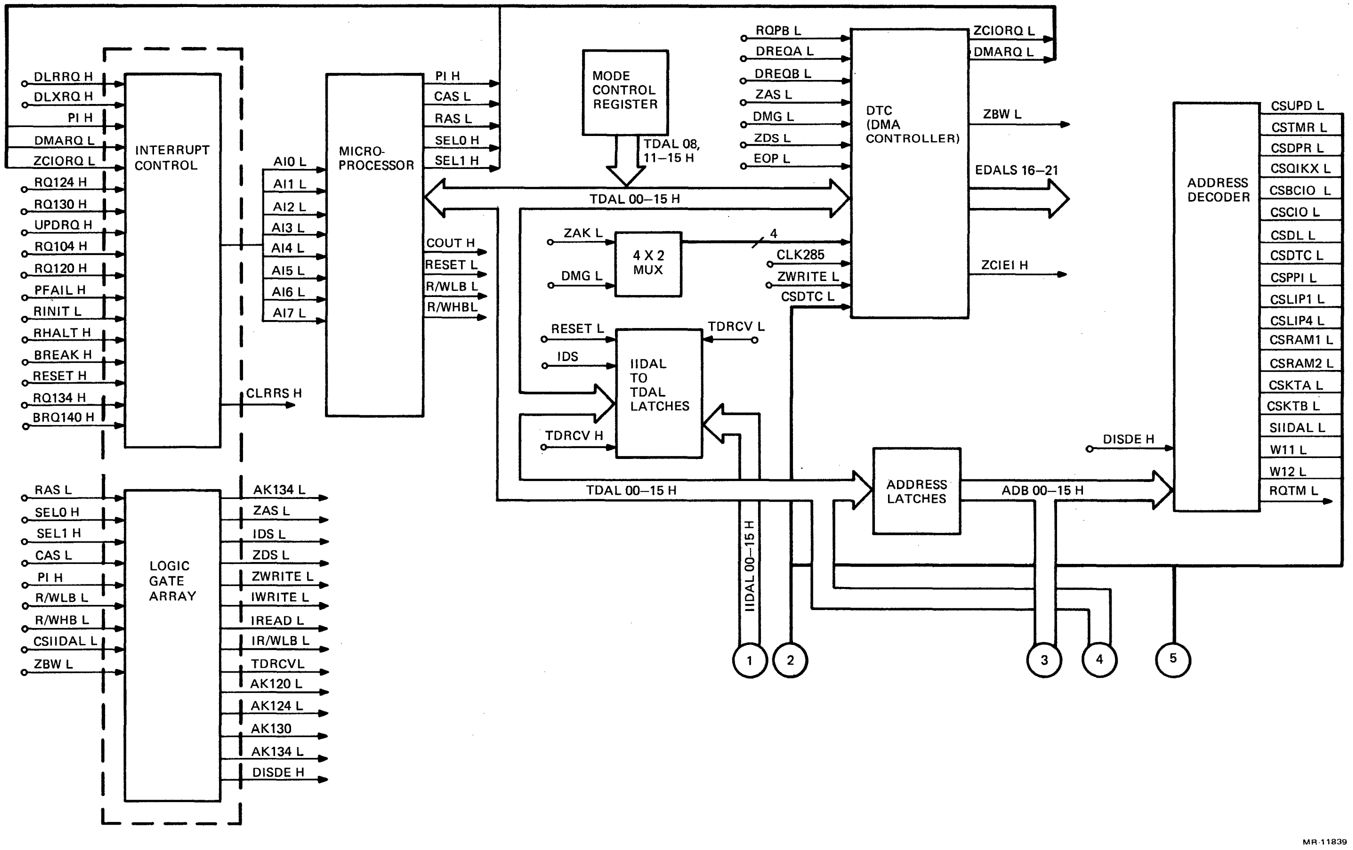


Figure 3-1 KXT11-CA SBC Functional Block Diagram (Sheet 1 of 3)

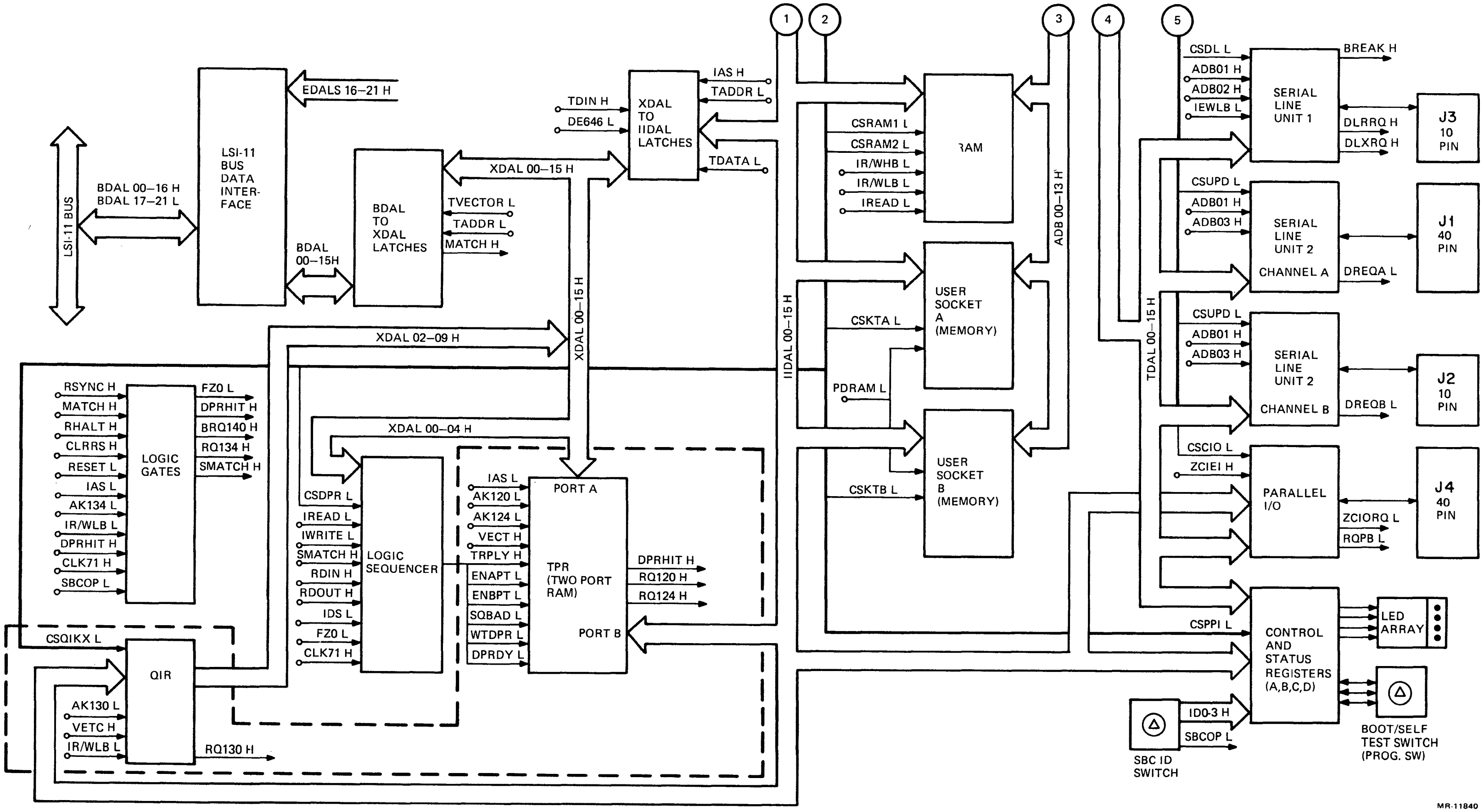


Figure 3-1 KXT11-CA SBC Functional Block Diagram (Sheet 2 of 3)

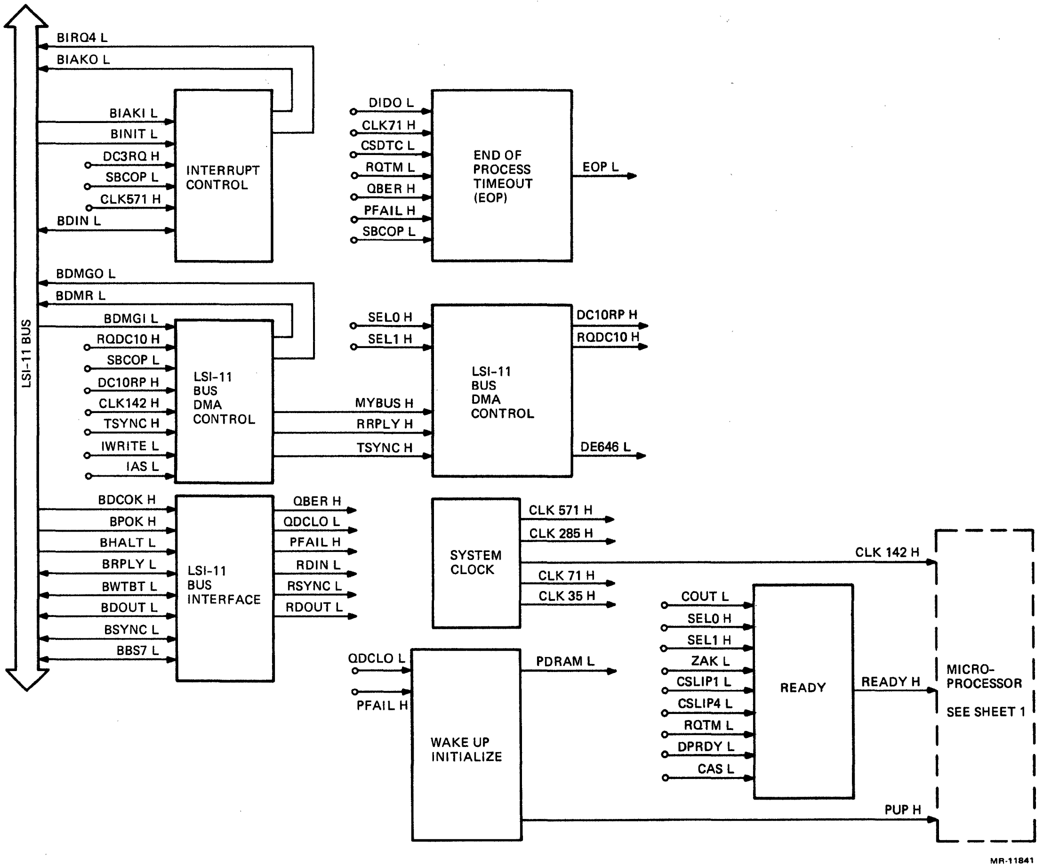


Figure 3-1 KXT11-CA SBC Functional Block Diagram (Sheet 3 of 3)

3.2 MICROPROCESSOR

The DCT11-AA microprocessor is contained within a 40-pin LSI chip and is shown in Figure 3-2. There are eight internal 16-bit general-purpose registers (R0–R7). R6 operates as a stack pointer (SP); R7 operates as the microprocessor program counter (PC). A special-purpose status register contains the current processor status word (PSW). The operating characteristics of the microprocessor are affected by the mode register which is discussed in detail in Paragraph 3.3.

3.2.1 Microprocessor Initialization

The microprocessor initializes the KXT11-CA SBC during the power-up sequence or when the RESET instruction is executed.

3.2.1.1 Power-Up Input (PUP) – The power-up (PUP) input goes from a high to a low transition approximately 47 ms after the 5 V power is first applied initiating the power-up sequence. The BCLR L output is then asserted. The SBC functions are reset as for the RESET instruction; however, both serial line units (SLU1 and -2), control status registers (CSR A, B, C and D) are cleared and reset. In the peripheral processor mode, after a delay, incoming Q-Bus signals BDCOK L and BPOK L are asserted, PUP H is negated, and the BCLR L output goes high. In the standalone mode, PUP H is negated by

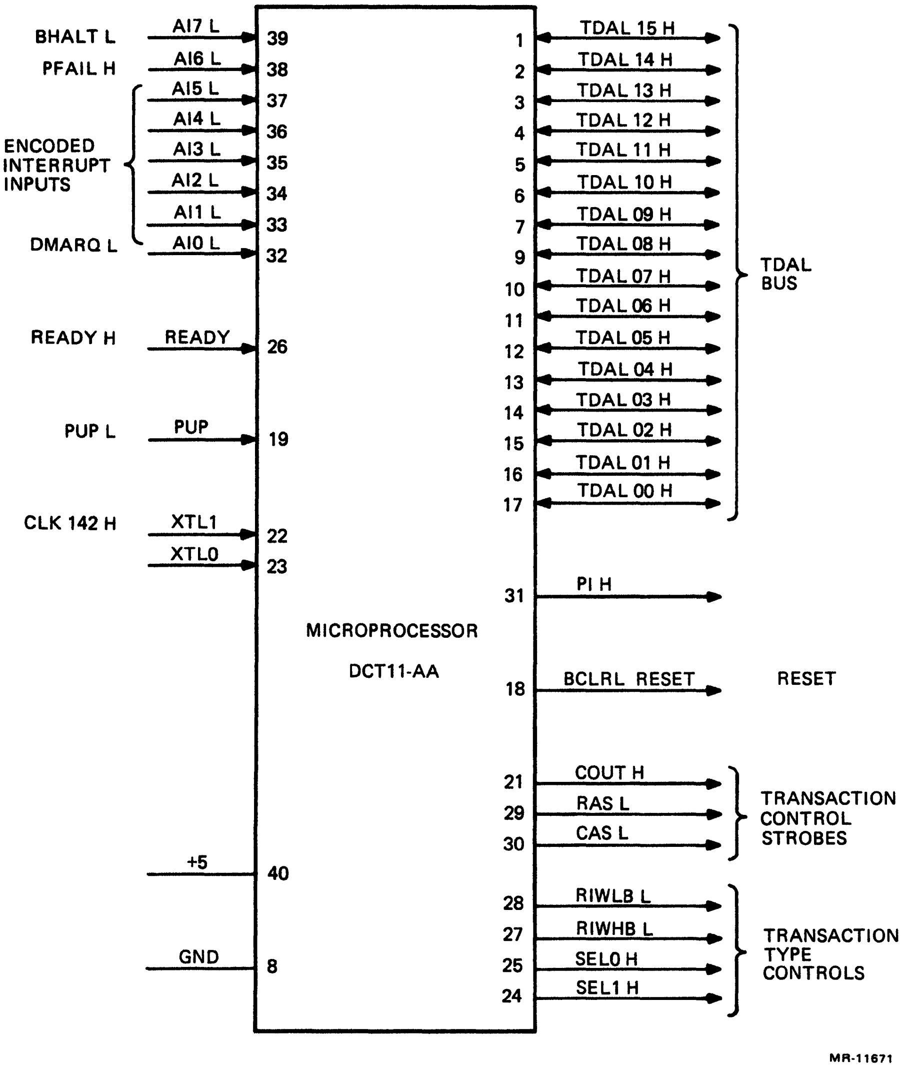


Figure 3-2 DCT11-AA Microprocessor

clearing the PUP flip-flop during the assertion of TPUP L. The microprocessor then performs 10 bus NOP transactions. The processor loads the starting address into the program counter (R7) and loads location 376 into the stack pointer (R6); the processor status word is set to 340. An assert priority in (ASPI) transaction is performed (PI H is asserted) to service any interrupts or DMA requests before the first instruction is fetched.

The PUP H input normally stays low for all operations. If PUP H is asserted high, the present transaction is terminated and the internal registers go to an undetermined state. The local TDAL bus, the interrupt inputs, and the microprocessor control signals will all go to an initial reset state.

3.2.1.2 Reset Instruction – The reset instruction asserts the BCLR L output. This clears or resets the control logic of the SBC to an initial state. The microprocessor loads the mode register from the TDAL bus with the mode register control data. The Q-Bus transceivers and local TDAL–IIDAL bus transceivers are disabled when BCLR L is asserted. All interrupts from the gate array (vector encode circuitry) to the T-11 are inhibited and any pending latched requests are cleared. All bits in KXT control status register (CSR) D are cleared.

The BCLR L output is then negated and an assert priority in (ASPI) transaction is performed to service any interrupts or DMA requests. The reset instruction does not change the PSW or any internal register values.

3.2.2 Microprocessor Control Signals

The microprocessor controls the functions of the KXT11-CA through the use of nine microprocessor control signals. A description of these signals and their functions follows. The RAS, CAS, PI, COUT, and BCLR are transaction control strobes used for logic transitions. The R/WLB L, R/WHB L, SEL0, and SEL1 are steady state logic signals used as "transaction type" control signals.

3.2.2.1 Row Address Strobe (RAS L) – The leading edge of the RAS L signal is used to acknowledge that the address is stable on the TDAL bus during read/write and fetch transactions. During interrupt transactions, the leading edge of the RAS L signal strobes the interrupt acknowledge data onto the TDAL 12–8 bus lines.

3.2.2.2 Column Address Strobe (CAS L) – The trailing edge of the CAS L signal is used to acknowledge that data on the TDAL bus lines during read and fetch transactions was read by the microprocessor. The trailing edge is also used to enable the microprocessor to read the interrupt inputs AI0 L to AI7 L and to initiate IAK, restart, power fail, or DMA transactions. For write transactions, the signal is used to acknowledge that microprocessor data will be removed after a specified time.

The leading edge of the signal is used to request that read data be placed on the TDAL bus and to strobe interrupt requests into latches that are read during the assertion of PI H.

3.2.2.3 Priority In (PI H) – The leading edge of the PI H signal is used to acknowledge that data on the TDAL bus lines during write transactions is stable. In addition, asserted PI H enables encoded information to be driven on the AI lines.

3.2.2.4 Read/Write (R/WHB L and R/WLB L) – The R/WHB L and R/WLB L signals control the read/write and fetch transactions by enabling T-11 reads and writes into and out of local RAM/ROM and I/O ports.

3.2.2.5 Select Output Flags (SEL0 H and SEL1 H) – The SEL0 H and SEL1 H signals indicate the transaction being performed. When both signals are low, a read, write, ASPI, or NOP transaction is selected. When both signals are high, a DMA transaction is selected. When SEL1 H is low and SEL0 H is high, the fetch transaction is selected. When SEL1 H is high and SEL0 H is low, an IAK transaction is being performed.

3.2.2.6 Bus Clear (BCLR L) – The BCLR L signal is used to reset the control logic. The signal is asserted during the power-up sequence and during the execution of a reset instruction only.

3.2.2.7 Clock Out (COUT H) – The COUT H signal is asserted once for every microcycle and is used to synchronize external signals with the T-11.

3.2.2.8 Clock (CLK 142 H) – The CLK 142 H input is a 7-MHz clock that is derived from the 28-MHz crystal oscillator. This clock input is used for the internal time base of the microprocessor and the source of the clock output (COUT). COUT is pulsed once for every microcycle. A microcycle can represent either three or four CLK 142 H input pulses depending on the type of transaction. The microprocessor will halt or stop when the CLK 142 H input is disabled.

3.2.2.9 Ready (READY H) – The READY input is normally high and will not interfere with microprocessor transactions. However, when the input pulsed, a single microcycle slip occurs during every bus transaction. When READY is clocked with COUT H, while RAS L is asserted, the microprocessor slips a microcycle every time the input is pulsed. This allows the microprocessor to be placed in an idle or wait state until a peripheral device has either received or asserted data on the bus.

3.2.3 Microprocessor Transactions

The microprocessor performs the following six types of transactions to support the instruction set, direct memory access, and the interrupt structure.

1. Fetch/read
2. Write (word or byte)
3. DMA
4. IAK/External/Internal
5. ASPI
6. Bus NOP

A normal fetch/read or IAK transaction requires either one or two microcycles; extended transactions can take multiple microcycles as required. The COUT H signal is asserted once for every microcycle. The transactions are used to transfer information and data via the TDAL bus which interconnects all local devices and connects them to the Q-Bus interface. A description of each transaction operation follows.

3.2.3.1 Fetch/Read – The fetch/read transaction is used either to fetch an instruction or read data for the microprocessor. The data may originate from the on-board memory I/O device. The microprocessor control signals for the transaction are shown in Figure 3-3. The R/WLB L and R/WHB L control signals are unasserted. The SEL0 H output is high, and the SEL1 H output is low for the fetch transaction; both of these outputs are low for the read transaction.

The following sequence of events takes place during a fetch/read transaction.

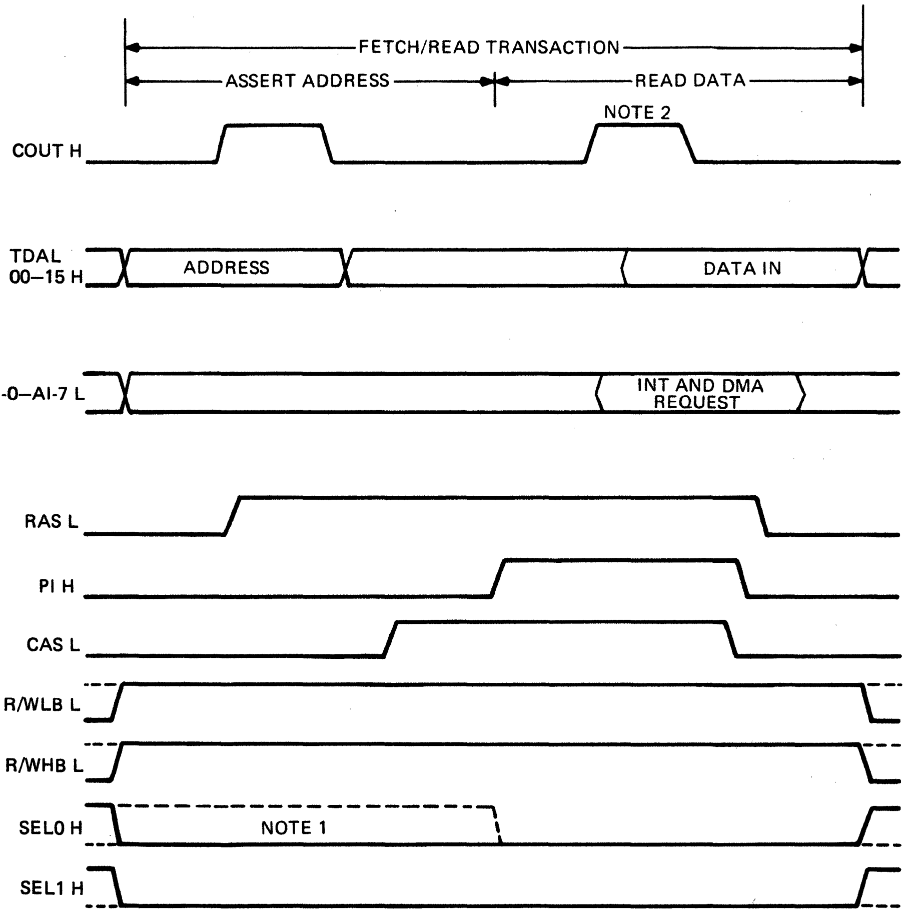
1. The microprocessor places the address onto the TDAL bus when the transaction is initiated and is latched into the memory address circuits by the assertion of IAS L (derived from RAS L).

2. The data is received on the TDAL bus after RAS L is received. The microprocessor accepts the data and asserts CAS L.

3. Interrupt and DMA requests are latched by CAS L, set up while PI H is asserted, and latched into the microprocessor when PI H is asserted.

NOTE

A write transaction is always preceded by a read transaction except when the microprocessor pushes onto the stack. Therefore, each write has at least four microcycles: assert address, read data, assert address, and write data.



NOTES:
1. SEL0 IS HIGH FOR FETCH TRANSACTIONS AND LOW FOR READ TRANSACTIONS.
2. TDAL BUS TRANSACTIONS CAN CAUSE THIS PORTION OF TIME TO SLIP UNTIL THE DEVICE RESPONDS OR TIME-OUT OCCURS.

MR-12462

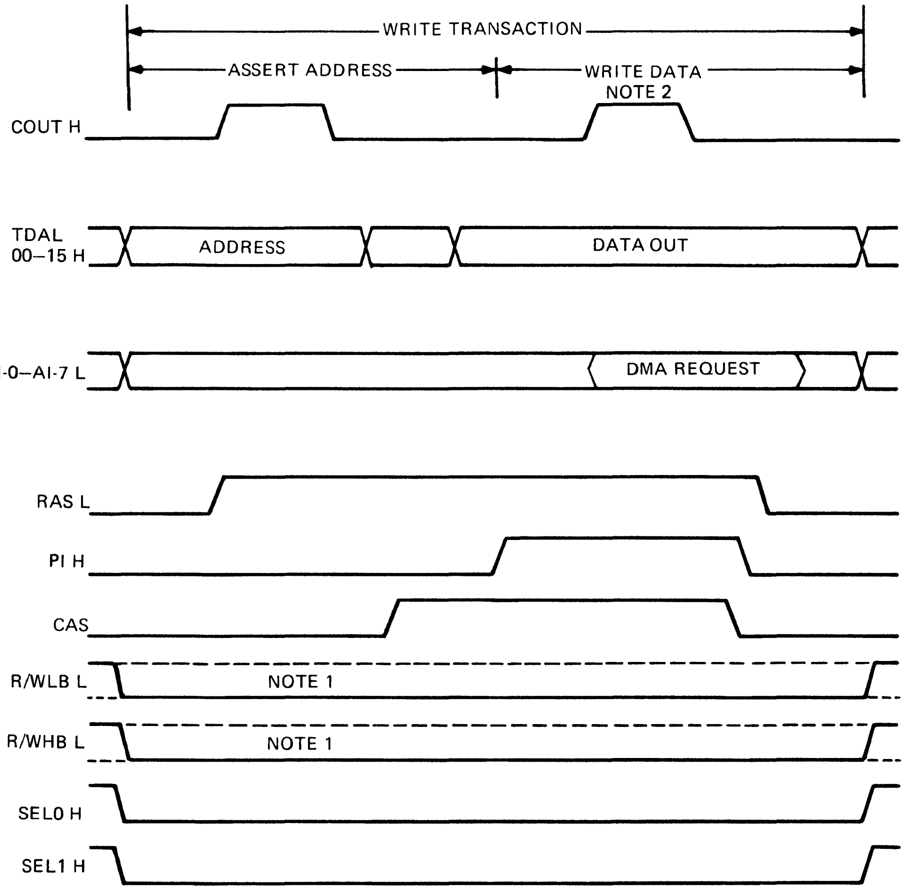
Figure 3-3 Read/Fetch Transaction

3.2.3.2 Write – The write transaction is used to write data from the microprocessor to memory or a local I/O device. The microprocessor control signals for the transaction are shown in Figure 3-4. The R/WLB L and R/WHB L control signals are asserted low when writing a word; when writing a byte, either the high or low byte signal is asserted. Both SEL0 and SEL1 control signals are asserted.

The following sequence of events take place during a write transaction.

1. The microprocessor places the address onto the TDAL bus, and the state of the read/write lines causes IWRITE L to assert. The address is latched into the memory address decoder by the assertion of RAS L.

2. When CAS is asserted, IWRITE L is unasserted for word transactions and left asserted for byte transactions.



NOTES:
1. R/WHB L OR R/WLB L CAN BE HIGH WHEN PERFORMING A WRITE BYTE TRANSACTION.
2. TDAL BUS TRANSACTIONS CAN CAUSE THIS PORTION OF THE TIME TO SLIP UNTIL THE DEVICE RESPONDS OR TIME-OUT OCCURS.

MR-12461

Figure 3-4 Write Transaction

3. The data is placed on the TDAL bus before PI H is asserted. The data is written into the addressed location when PI H is asserted.

4. The DMA requests are detected while PI H is asserted; they are latched into the microprocessor when PI H is unasserted. No other interrupts are read by the microprocessor during write transactions.

3.2.3.3 IAK/Internal/External – If an interrupt request was detected during a previous read transaction, the microprocessor initiates an IAK transaction as shown in Figure 3-5. The CAS L, R/WHB L and R/WLB L control signals are asserted; PI H and SEL0 H are unasserted for the IAK transaction. TDAL bits 12–8 output a code representing the acknowledged input and are used to reset the interrupt request logic. The T-11 chip can service both local and external interrupts.

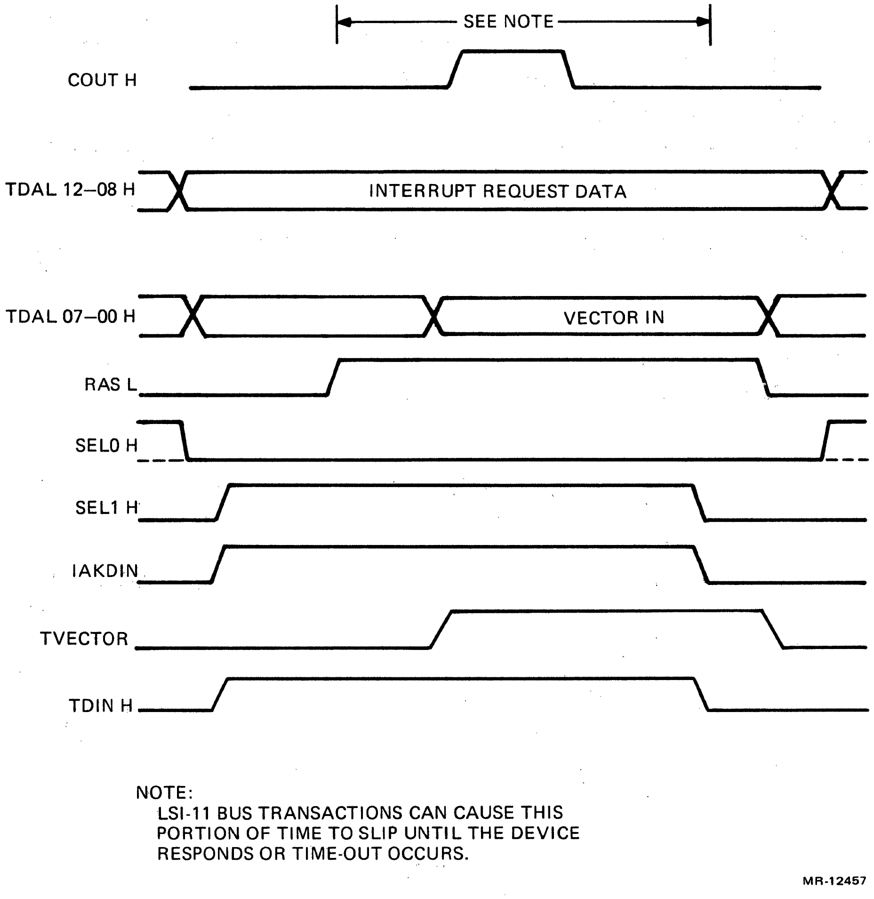


Figure 3-5 IAK Transaction

For local interrupts, TDAL bits 7–0 are ignored because the vector address is located in the microprocessor. For external (DTC, PI/O) interrupts, the vector address is read from TDAL bus bits 7–2. TDAL 12 (VECT L) is set low for external IAK transactions and commands the control logic to initiate an external vector IAK transaction. TDIN H is asserted for the transaction, TVECTOR L acknowledges the interrupt. The requesting device places the vector address on low byte of the Q-Bus and asserts B RPLY L. The microprocessor asserts TDIN H and accepts the vector. Software programmable vectors are read into the T-11 when VEC L is asserted.

3.2.3.4 DMA – The DMA request is read during a previous transaction. The microprocessor will acknowledge the request by tristating the TDAL bus as shown in Figure 3-6. The SEL0 H and SEL1 H outputs are asserted to indicate that the bus mastership has been relinquished to an external device (DTC). The DTC is the only device that can acquire local bus mastership from the T-11. The transaction will continue with no interruptions until the DMA transfer is completed. The microprocessor will then unassert the SEL1 H control output to indicate that the T-11 is resuming bus mastership. The unassertion of SEL0 H will follow if the next transaction is not a fetch.

3.2.3.5 ASPI – The assert priority in (ASPI) transaction is used by the reset and wait instructions or during the power-up sequence as shown in Figure 3-7. The CAS L and PI H outputs are asserted, enabling the microprocessor to recognize and latch any interrupts or DMA requests. The R/WHB L and R/WLB L outputs are unasserted and the SEL0 H, SEL1 H, and RAS L outputs are negated for the transaction.

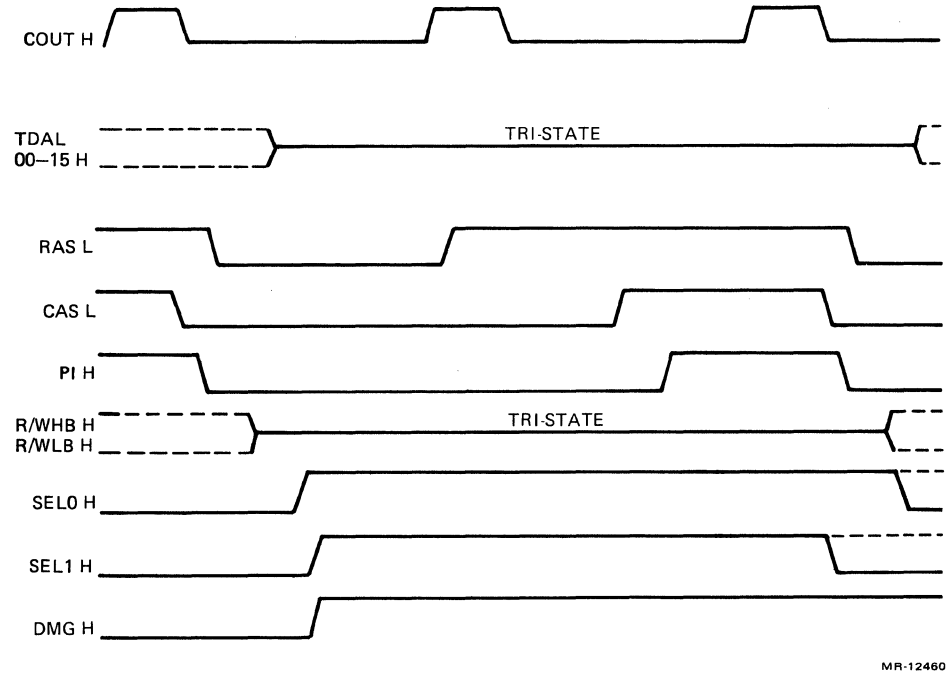


Figure 3-6 DMA Transaction

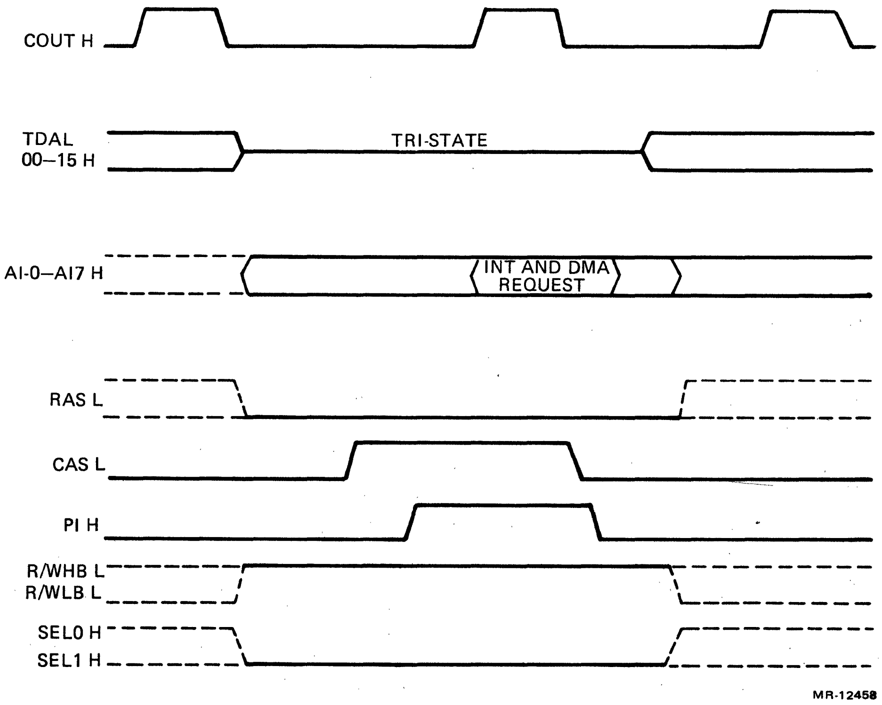


Figure 3-7 ASPI Transaction

3.2.3.6 NOP – The bus NOP transaction performs no operation and is used during the power-up sequence or if the programmer intentionally introduces a delay into the program. The AI0 L–AI7 L inputs are tristated to prevent interrupts. The R/WHB L and R/WLB L outputs are asserted, and the SEL0 H and SEL1 H outputs are unasserted low. The RAS L, CAS L, and PI H control strobes are inhibited during the transaction as shown in Figure 3-8.

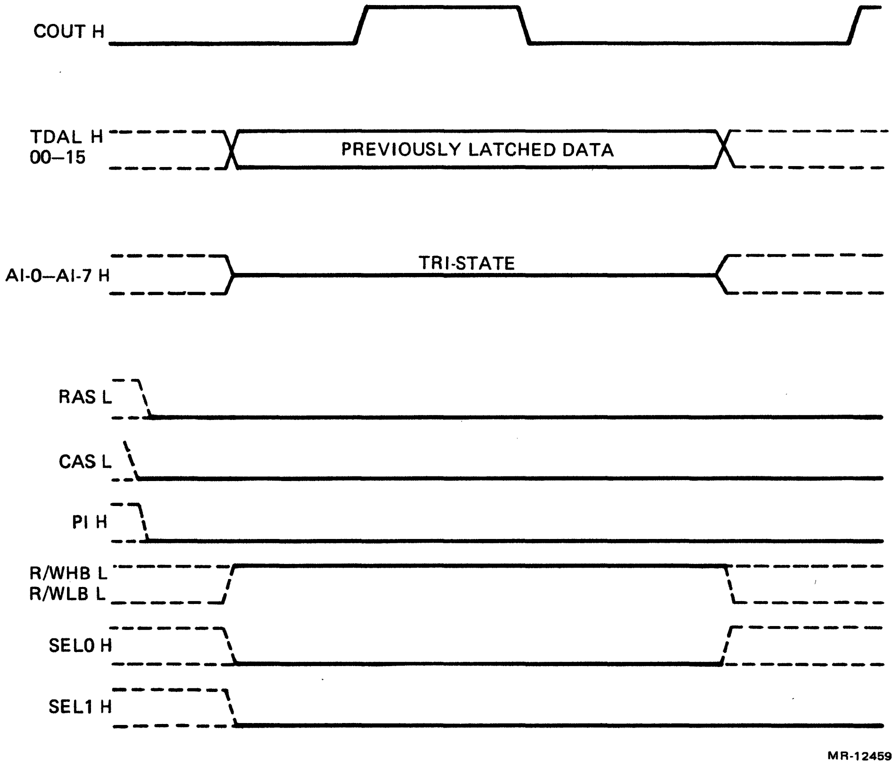


Figure 3-8 NOP Transaction

3.3 MODE REGISTER

The mode register is used to define the operating mode of the T-11 microprocessor. The 16-bit mode register is written into from the TDAL 0–15 data lines during a power-up sequence or when a reset instruction is executed. During this time, the BCLR L output is asserted and the mode register is loaded. The mode register logic (Figure 3-9) has five tristate drivers that are enabled. TDAL bits 8 and 11 are factory set to force the microprocessor to operate in the following mode.

1. The microprocessor clock mode is selected. The microprocessor pulses the COUT output once for every four XTL1 input pulses during DMA and interrupt transactions. For all other transactions, it pulses the COUT output once for every three XTL1 input pulses.

2. The standard microcycle mode is selected. This mode uses four XTL1 input periods for DMA and interrupt transactions and three XTL1 input periods for all other transactions.

3. The normal read/write mode is selected. This mode sets the read/write control lines (R/WLB L and R/WHB L) prior to the assertion of –RAS and remains valid after the negation of CAS L.

4. The static memory mode is selected, and therefore, no dynamic memory chips may be installed on the board. The refresh function is disabled.

5. The memory addressing is limited to 64 Kb.

6. The bus has 16 bits.

7. The user mode is selected. This mode performs transactions with no automatic test of the processor status word.

The status of TDAL bits 13–15 are factory set and determine the start address for the microprocessor. The start address is the location of the first fetch after power-up.

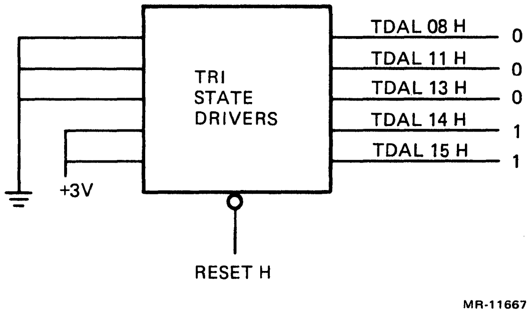


Figure 3-9 Mode Control Register

3.4 CONTROL STATUS REGISTERS (KXTCSR)

The local processor has four I/O page addressable registers for controlling and monitoring the SBC system environment. The following represents an overview of each register. More specific register bit detail and usage can be found in Chapter 5.

3.4.1 KXTCSR A

SBC control status register A is an 8-bit read/write word or low byte addressable register providing synchronous/asynchronous clock control bits, party line receiver enable bit, modem control bits, diagnostic PROM control bit, real-time clock (RTC) enable bit, and programmable clock-interrupt control bits.

3.4.2 KXTCSR B

SBC control status register B is an 8-bit read only low byte or word-addressable register providing status for synchronous/asynchronous modem line, memory map configuration, and the boot/selftest switch selection.

3.4.3 KXTCSR C

SBC control status register C is an 8-bit read only (RO) low byte or word-addressable register providing status for SBC ID switch setting and four programmable red LEDs for error reporting.

3.4.4 KXTCSR D

SBC control status register D is a 16-bit read/write word-addressable register providing status and control for nonexistent memory (NXM) flag, Q-Bus interrupt register (QIR), full flag, QIR interrupt enables, Q-Bus reset occurred flag with trap enable, BHALT trap enable control, two-port register (TPR) restart occurred flag bit, TPR enable bit, TPR interrupt enable bits, and TPR interrupt status flags.

3.5 SYSTEM CLOCKS

The KXT11-CA SBC uses two crystal oscillators: one 28-MHz and one 19.660-MHz. The 28-MHz oscillator is used as the basic time base reference for the microprocessor. The 19.660-MHz oscillator is used as the basic time reference for the I/O ports.

The 28-MHz oscillator (Y1) output (Figure 3-10) goes to the divide-by-7 clock control logic and to the binary counters. During the power-up sequence, TPUP L asserts to load the counter with data. The 28-MHz output is divided by seven, providing a 4-MHz output (CLK250 H) that is used in the counter timer parallel I/O chip and SLU2. The 28-MHz output is also divided by two, and the (14-MHz) output (CLK71 H) goes to the logic sequencer. This 14-MHz output is further divided by 2, 4, and 8. The divide-by-2 (7-MHz) output (CLK 142 H) goes to the T-11 microprocessor and the Q-Bus DMA request (DC0010) chip. The divide-by-4 output, a 3.5-MHz output (CLK285 H), goes to the DTC. The divide-by-8 output, a 1.75-MHz output (CLK571 H), goes to the DC0003 interrupt control chip. During the KXT11-CA power-up cycle, the asserted TPUP L signal clears the following counters: divide-by-7, divide-by-4, and divide-by-8. Asserted TPUP L also presets both divide-by-2 counters.

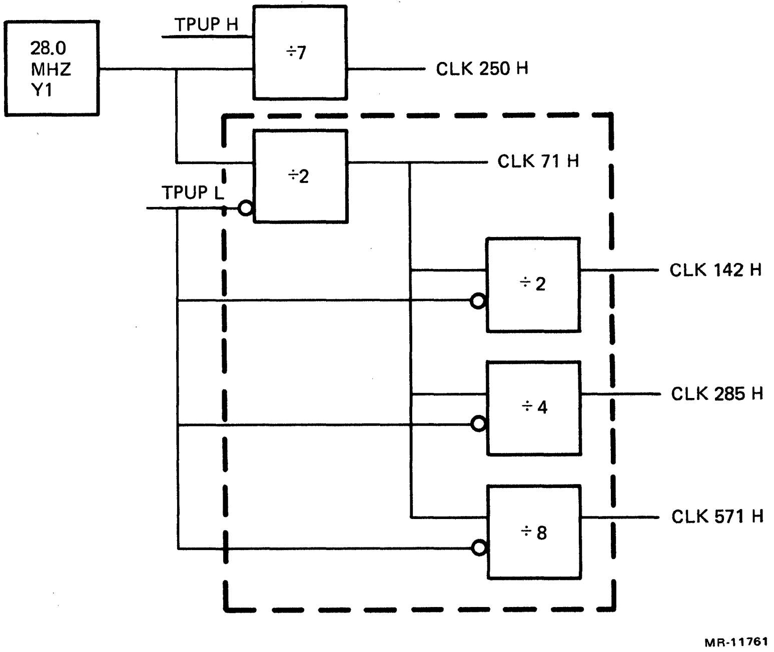


Figure 3-10 28-MHz System Clocks

The 19.660-MHz oscillator (Y2) output (Figure 3-11) goes to a dual 4-bit binary counter. These binary counters are always enabled. The output of Y2 is divided by two, providing a 9.83-MHz output (CLK101 H) that is used to drive the output of the programmable baud rate generator. The 19.06608-MHz is divided by 32, providing a 614.4 KHz output (CLK016 H) used to drive SLU1 driver and 12 V charge pump.

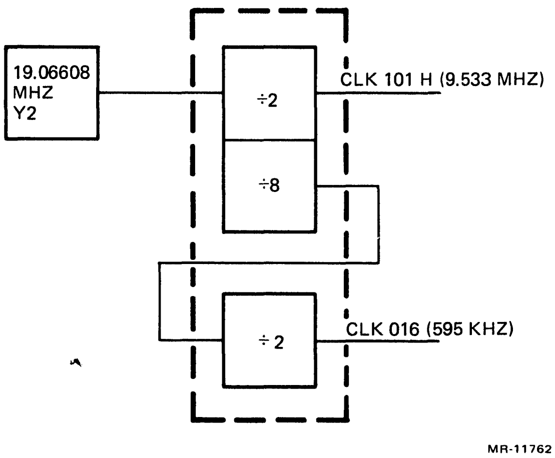


Figure 3-11 19.066-MHz System Clock

3.6 Q-BUS INTERRUPT PROTOCOL CHIP (DC003)

The DC003 protocol logic chip (Figure 3-12) is used to generate a level 4 interrupt on the Q-Bus with a programmable vector address generated by the KXT11-CA SBC. When an interrupt request is pending, the Q-Bus interrupt register (QIR) sends a request to the DC003 chip. The DC003 request signal, DC3RQ H, asserts, enabling the bus interrupt request line (BIRQ4 L) to assert on the next low-to-high transition of the 1.75-MHz clock (CLK571 H). The bus arbiter acknowledges the assertion of BIRQ4 L by asserting RDIN L and then asserts the BIAKI L line. This BIAKI L signal is daisy chained through each board. The BIAKI L line of one board is the BIAKO L line from the electrically closer board, to the arbiter in the daisy chain. The first board with an interrupt request (BIRQ4 L line asserted) captures the BIAKI L and inhibits the remaining boards in the daisy chain from receiving the IAK by keeping BIAKO L unasserted.

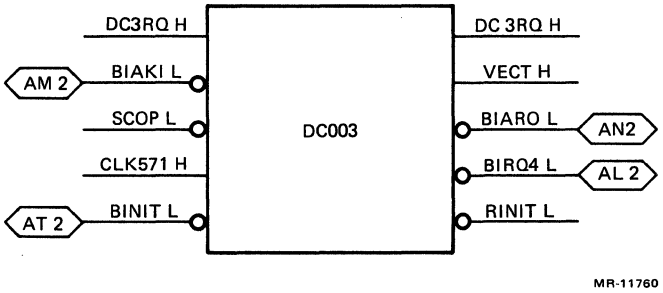


Figure 3-12 Q-Bus Interrupt Chip (DC003)

After capturing the bus interrupt acknowledge, VECT H becomes asserted. This enables the SBC's Q-Bus transceivers to drive the interrupt vector onto the Q-Bus. When the interrupt request cycle is completed, the DC3RQ L line becomes unasserted high. This causes the BIRQ L line to become unasserted and the bus arbiter unasserts lines RDIN L and BIAKO L. The DC003 protocol is used only during Q-Bus interrupt operations. During standalone mode, SCOP L is asserted, inhibiting all DC003 outputs except for RINIT L. The KXT11-CA SBC can then detect and respond to Q-Bus resets (looks at state of BINIT L).

When the KXT11-CA is operated in the peripheral processor mode and the arbiter asserts BINIT L, all lines driven by the DC0003 become unasserted except for output RINIT L. This line asserts, causing the SBC to perform a trap to location 24, if enabled.

3.7 Q-BUS DMA PROTOCOL CHIP (DC010)

The DC010 DMA protocol (Figure 3-13) is used by the SBC's DTC (DMA controller chip) to gain control of the Q-Bus for read/write DMA transfers. The DTC sequence is operated by a 7-MHz clock rate (CLK142 H). Read/write DMA transactions are initiated by asserting the request line RQDC10 H. The asserted RQDC10 H combines with the unasserted SBCOP L to assert the bus direct memory request (BDMR L) line. The Q-Bus arbiter responds to the direct memory request by asserting the bus direct memory grant line (BDMGI L), which causes the direct memory grant output (BDMGO L) to the unasserted state. This ensures that only the electrically closest (to the arbiter) requesting SBC gains control of the bus. BDMGI L is daisy chained through each board in the daisy chain. The BDMGI L line of one board is the BDMGO L line from the electrically closer board in the daisy chain. The first board requesting a direct memory grant inhibits the remaining daisy-chained boards from receiving a grant by keeping BDMR L unasserted.

When the arbiter's current operation cycle is completed, RSYNC H becomes unasserted. This enables the SBC to assert MYBUS H and send a control signal to the Q-Bus. The DTC has now completed the acquisition of the Q-Bus and is ready to do a read or write cycle.

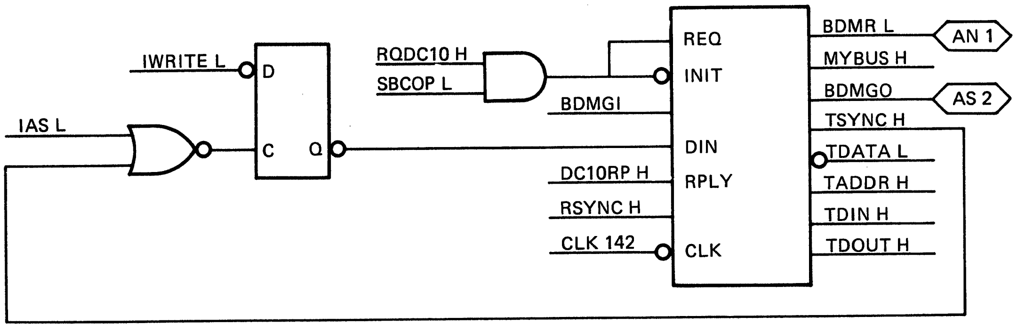


Figure 3-13 Q-Bus DMA Protocol Chip (DC010)

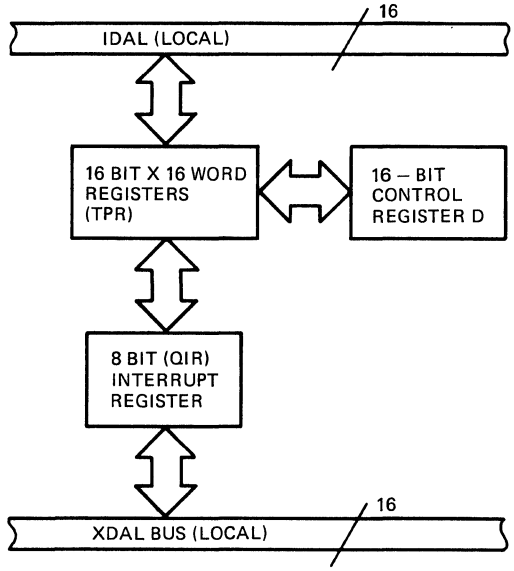
The following occurs when a Q-Bus write cycle is requested. During this transaction the SBC is Q-Bus master and the arbiter is Q-Bus slave. Both IAS L and IWRITE L are asserted in sequence, toggling the flip-flop, thereby supplying a low to DC010 input RDIN. The DC010 asserts the transmit address signal TADDR H used by the SBC to latch the address onto the Q-Bus. After latching the outgoing address, the DC010 asserts transmit sync signal TSYNC H used by the slave device to latch in the address. Next, the DC010 latches outgoing data onto the Q-Bus by asserting TDATA L.

The DC010 now asserts TDOUT H and is used by the slave device to latch in the data from the Q-Bus. After data is latched into the bus processor, it responds by asserting BRPLY L, which is then received by the DC010 as an asserted reply (DC10RP H). This ends the write cycle, causing TDOUT H, TSYNC H to unassert and begin the next cycle.

When a Q-Bus read cycle is requested, the Q-Bus slave (arbiter during this interval) first latches the address asserted by the SBC when TSYNC H is asserted. For read cycles IWRITE L is unasserted. DC010 requests data from the slave device by asserting TDIN H. After receiving TDIN H the slave drives addressed data onto the Q-Bus and asserts BRPLY L, indicating that data on the Q-Bus is valid. The asserted BRPLY L combines with asserted MYBUS H in the Q-Bus DMA control portion of the DC340 to provide the DC010 with an asserted DC10RP H input. Data is then latched into the SBC when TDIN H becomes unasserted during a high-to-low transition.

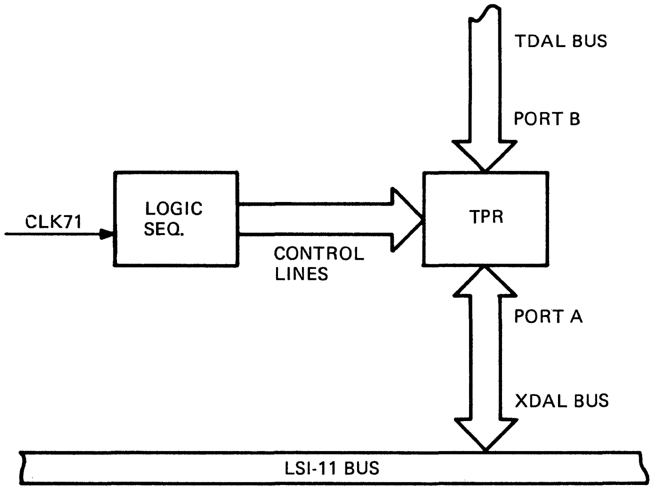
3.8 Q-BUS INTERFACE GATE ARRAY (DC340)

The DC340 gate array contains a 16-bit × 16-word two-port register (TPR) which includes a 16-bit system control register (TPR Register 0) that is a read only from the local bus and a wwrite only from the Q-Bus. In addition, the DC340 contains the QIR, an 8-bit vector interrupt register. Figure 3-14 shows the DC340 logic. The DC340 is intended to operate with a state machine that provides the two-port RAM write access control arbitration between local bus and Q-Bus access. (See Figure 3-15.) The local bus consists of the bidirectional TDAL00 H–TDAL15 H lines. The Q-Bus consists of the bidirectional XDAL00 H –XDAL15 H lines.


16
IDAL (LOCAL)
16 BIT X 16 WORD REGISTERS (TPR)
16 – BIT CONTROL REGISTER D
8 BIT (QIR) INTERRUPT REGISTER
16
XDAL BUS (LOCAL)

MR-11758

Figure 3-14 Q-Bus Interface Gate Array Simplified Block Diagram (DC340)



MR-11756

Figure 3-15 TPR and Logic Simplified Block Diagram

3.8.1 Two-Port RAM (TPR)

The two-port RAM is a 16-bit × 16-word file. (See Figure 3-15.) Concurrent reads can occur between the Q-Bus and local TDAL bus; concurrent read/write combinations from both buses can occur. The local TDAL bus has read/write priority over the Q-Bus, enabling some T-11 read-modify-write operations. This word file is Q-Bus addressable and has an address that changes with the SBC system ID switch setting. Q-Bus writes to the fifth, ninth or thirteenth words cause flags to be set in control register D and/or interrupts to the T-11 through vectors 120, 124, and 134 on level 5. A priority channel is provided by the first word (0), so that when written from the Q-Bus, a T-11 restart trap is generated and a flag bit in register D is set. All words are addressable from both the Q-Bus and the T-11. Not all words are R/W from both buses. The TPR state machine diagram is on page D-19.

Read/write cycles (IICYC) are performed using TPR port B and local TDAL bus. To do a local read (IIRD cycle) a logic sequencer must assert two-port ready input enable line DPRDY L. Then the logic sequencer asserts the B port input enable line ENBPT L. These two asserted outputs enable the TDAL bus access to the TPR through port B.

To do a local write (IIWT) the logic sequencer must assert two-port ready output DPRDY L. Then the logic sequencer asserts the write into two-port RAM enable line WTDPR L. These two asserted inputs enable the TDAL bus data to be driven into the TPR through port B.

The Q-Bus arbiter can perform read/write operations in/out of the TPR using port A. TPR operations are enabled by setting TDAL bit 6. Prior to TDAL bit 6 being set, the arbiter is limited to writing only into the first TPR register. In addition, control status register D (KXCSRD) and the second, sixth, and tenth words are Q-Bus read only.

Attempted writes from the Q-Bus processor into any other word causes a timeout to occur. Attempted Q-Bus processor reads of any readable TPR word asserts FZ0 H and forces all TPR word contents to be read by the Q-Bus processor as zeroes. Asserted FZ0 L is used as an input to the logic sequencer that inhibits Q-Bus read sequencing. Until the TPR is fully initialized, FZ0 L is asserted, inhibiting any TPR Q-Bus read operations. Write operations from the Q-Bus are limited to TPR register 0. In addition, the local TDAL bus can do read/write TPR operations. When the TPR is fully initialized, TDAL 6 acts as the TPR Q-Bus read/write enable (FZ0 L becomes unasserted).

During a Q-Bus read cycle, the arbiter selects the TPR by asserting MATCH H with RSYNC H. These two assertions combine and assert the TPR ready line DPRHIT H. This high is clocked through a flip-flop and becomes logic sequencer asserted SMATCH H. Next, the Q-Bus processor asserts RDIN H. The logic sequencer receives RDIN H and asserts both ENAPT L and then TRPLY L. These two assertions enable port A to drive selected TPR data onto the BDAL bus via internal XDAL bus. After data is transferred either RDIN H or SMATCH H becomes unasserted ending the read cycle.

During a Q-Bus write cycle, the MATCH H and RSYNC H lines are asserted when the Q-Bus processor addresses the TPR. To strobe the Q-Bus address into the TPR, the logic sequencer now asserts SQBAD L. Next, WTPDR L becomes asserted, enabling the Q-Bus arbiter to write into the TPR file. The data is transferred via port A into the TPR when input strobe TRPLY L asserts. The selected Q-Bus data now resides in the TPR file.

3.8.2 Q-Bus Interrupt Register (QIR)

The QIR is an 8-bit, word addressable, write only vector register (Figure 3-16) used to signal the Q-Bus arbiter (with a level 4 interrupt request) that it requires attention or that some task has been completed. The local microprocessor can write vector values between 0 and 1774 into this register using the local TDAL bus. The QIR is not initialized on power-up, and is only cleared by a local reset or by a Q-Bus interrupt acknowledge. The register is updated with each T-11 write to the register. Bit 13 of register KXTCSR D must be set to enable the QIR. (See Chapter 5 for register D details.)

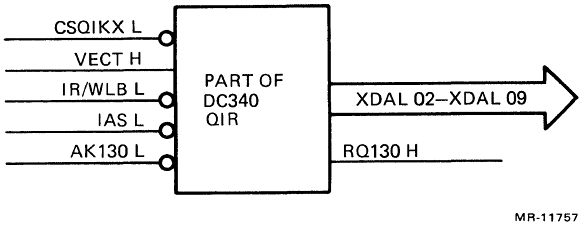


Figure 3-16 Q-Bus Vector Interrupt Register

The local T-11 processor initiates QIR operations by asserting chip select line CSQIKX L when address strobe IAS L is asserted during a T-11 write cycle (IR/WLB L asserted). The interrupt vector data on TDAL 02–09 is then latched into the QIR. This fills the QIR causing DC3RQ H to assert. The DC003 interrupt control chip receives asserted DC3RQ H and responds by asserting BIRQ4 L on the Q-Bus. After receiving the Q-Bus BIAKI L assertion, the DC003 asserts the TVECT H line. The DC340 drives the vector onto the XDAL bus 02–09 lines.

The Q-Bus arbiter now latches in the vector and also signals the DC003 chip that the cycle is completed. This causes the TVECT H line to unassert. The high-to-low transition of the TVECT H line causes the RQ130 L local interrupt line to assert. The interrupt vector (RQ130 H) line generates a local T-11 interrupt through internal vector 130. This local interrupt is read into the T-11 and then is acknowledged by the assertion of AK130 L. This interrupt signals the T-11 that the previous Q-Bus interrupt request was acknowledged. The SBC interrupt is now ready to be serviced.

A local reset will flush the request, interrupt enable, and the vector 130 interrupt.

3.8.3 Logic Sequencer

The logic sequencer is used to control both local and Q-Bus arbitration for the two-port RAM. The logic sequencer runs at a 14-MHz clock rate (see Figure 3-15). Operations are initialized during the initial power-up interval when PUP H is asserted.

After power-up logic sequencer operations are unaffected by local or Q-Bus initializations. The sequencer controls the two possible TPR cycles: a local bus cycle (IICYC) and a Q-Bus cycle (QBCYC). Each of these cycles is further divided into read or write cycles. The state diagram for both TPR cycles is shown in Figure 3-17. A detailed state diagram is on page D-19. Input signals controlling local bus cycles (IICYC) are CSDPR L, IREAD L, IWRITE L, SMATCH H, and IDS L. The input signals controlling Q-Bus cycles (QBCYC) are SMATCH H, RDIN H, RDOUT H, FZ0 L, and ADB01 H–ADB04 H.

A brief description of both read and write TPR operations for local and Q-Bus cycles follows.

The logic sequencer remains in the initial state until either a local or Q-Bus request line becomes asserted; the CSDPR L line becomes asserted for a local cycle and the SMATCH L line becomes asserted for a Q-Bus cycle. The logic sequencer then enters the ARBB state to arbitrate between local and Q-Bus cycle requests.

When a local cycle (IICYC) is requested the IICYC state determines if a read or a write is requested. For a write cycle the IWRITE L input asserts; enabling the DPR ready output DPRDY L to assert. The next logic sequencer input needed is the data strobe assertion IDS L. After IDS L asserts, the logic sequencer enters the ITWT state and asserts two outputs: write WTDPR L and the two-port ready DPRDY L. During this interval a write into the DPR RAM occurs. The next CLK71 transition unasserts WTDPR L. The sequencer changes state when data strobe IDS L unasserts. The logic sequencer then returns to the initial state.

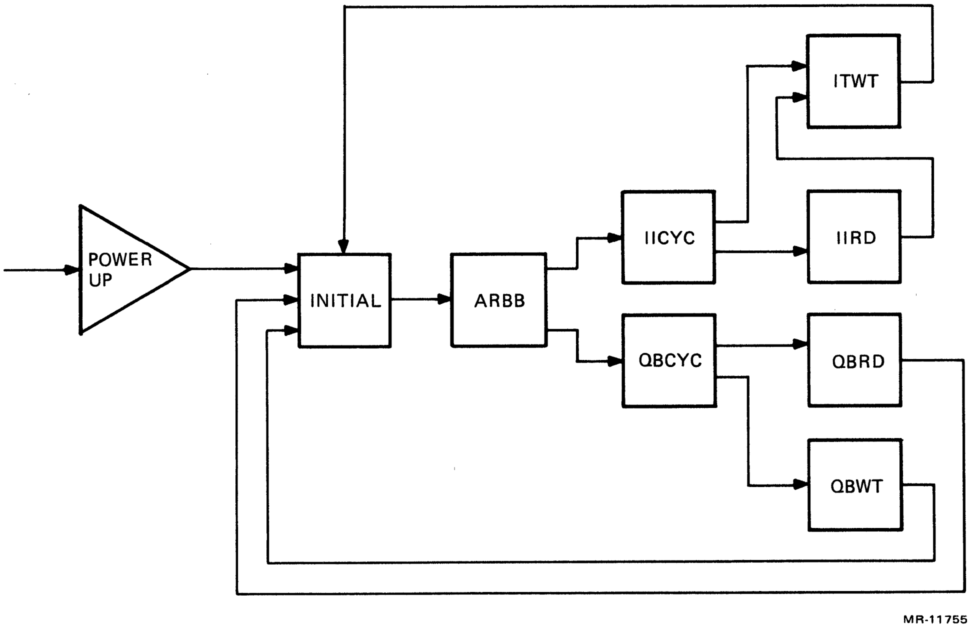


Figure 3-17 Logic Sequencer Functional Diagram

When a local read is requested the sequencer enters IIRD state when IREAD L asserts. Then, on CLK71 transition, the two-port ready signal DPRDY L asserts, enabling port B line ENBPT L to assert. The data is read during this interval and ENBPTL is asserted until IREAD L becomes unasserted. When this occurs the sequencer forces both DPRDY L and ENBPT L outputs to become unasserted. The sequencer goes through three NOP states for syncing purposes when a T-11 read/write cycle is done, and then enters the ITWT state checking for a write (IWRITE) request. If no write request is detected, the sequencer returns to the initial state.

A Q-Bus read occurs if SMATCH H and RDIN H input lines become asserted. The sequencer enters the QBCYC state and output line enable port A (ENAPT L) becomes asserted. On the next clock transition input, the logic sequencer output – transmit reply, TRPLY L – is asserted. The logic sequencer enters the QBRD state, a Q-Bus read occurs through port A, and the sequencer output, TRPLY L, remains asserted until input signal SMATCH H or RDIN H become unasserted. When this occurs the sequencer unasserts both ENAPT L and TRPLY L outputs. The sequencer then ensures all its outputs are unasserted and returns to the initial state.

A Q-Bus write occurs if SMATCH H and RDOUT H lines become asserted. The logic sequencer enters the QBCYC state and output select Q-Bus address, SQBAD L, is asserted, enabling the address selected. Next, a check of the address lines ADB01–04 ensures that Q-Bus writes are not loaded into read only registers. Another check ensures that the FZ0 L line is unasserted. This line is asserted by the DC339 logic during initial SBC power-up. After the check is complete, the logic sequencer enters the QBWT state, causing the write two-port RAM line, WTPDR L, to become asserted, enabling the transfer of data into the two-port RAM. On the next CLK71 transition, the reply signal TRPLY L output asserts and latches the XDAL data into the TPR registers. The next transition of CLK71 forces WTDPR L to unassert. The sequencer waits until either SMATCH H or RDIN H is unasserted by the Q-Bus device. Either forces all sequencer outputs to the unasserted state. The sequencer now returns to the initial state.

3.9 INTERRUPT CONTROL GATE ARRAY (DC339)

The DC339 chip is an 84-pin gate array used to provide a compatible interface between the T-11 microprocessor and the direct memory controller (DTC). Figure 3-18 shows the DC339 input and output signals.

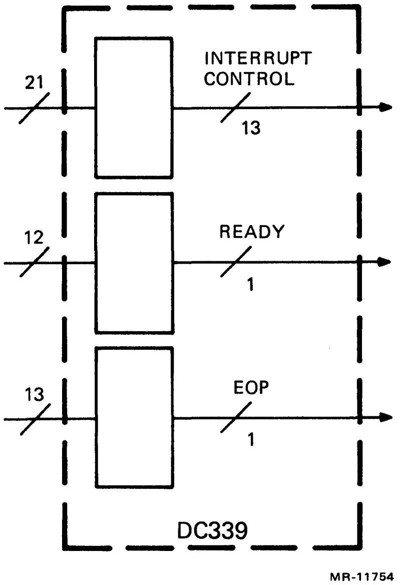


Figure 3-18 Interrupt Control Signals

3.10 INTERRUPT CONTROL

The interrupt control logic provides asynchronous wait state (READY H); interrupt and exception handling; address, read and write strobes; bus DMA arbitration control; and external/internal vector cycles.

3.10.1 Interrupt Hardware

The KXT11-CA interrupt control design (Figure 3-19) includes the following elements.

1. Seven interrupt lines that are latched.

a.	UPDRQ	SLU2 interrupt request
b.	RTCRQ	Real-Time clock 50- or 60-Hz interrupt request
c.	RQ 134	Two-Port RAM request
d.	RQ 130	Q-Bus interrupt register request
e.	RQ 120	Two-Port request
f.	RQ 124	Two-Port RAM request
g.	RQ 104	Timer counter register request

2. A logic circuit containing five inputs that latches two lines.

a.	RHALT	Q-Bus halt, nonmaskable level 7 interrupt
b.	PFAIL	Q-Bus power-fail, nonmaskable level 7 interrupt
c.	QBRESET	Q-Bus reset, nonmaskable level 7 interrupt
d.	BREAK	Console user selectable nonmaskable level 7 interrupt
e.	RINIT	Q-Bus initiate interrupt request

3. Thirteen interrupt synchronizing latches, that latch the following signals.

   a. Outputs of the seven latches described previously
   b. Two outputs from the logic circuit described previously
   c. Three interrupt signals:

      (1) ZCIORQ Counter PI/O timer interrupt request
      (2) DLRRQ Console receiver interrupt request
      (3) DLXRQ Console transmitter interrupt request

   d. DMARQ Direct memory request interrupt

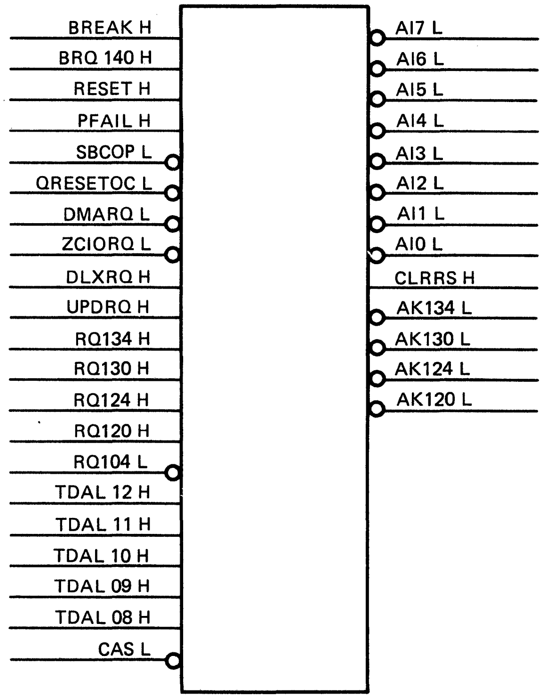


Figure 3-19 Interrupt Control Simplified Block

The operation of the KXT11-CA interrupt control centers around eight microprocessor input lines, AI0 to AI7, driven by interrupt signals, either directly or indirectly, through the DC339 gate array.

- AI0 is the DMA request line connected directly to DMRQ.
- AI1 to AI4 are driven by the output of the interrupt encoder to request maskable interrupts.
- AI5 is driven by the VEC gate which detects the presence of an externally generated vector on the outputs of the interrupt encoder. It calls for a vector read transaction from the bus.
- AI6 is ORed between the power-fail and Q-Bus input lines to force a power fail interrupt.
- AI7 is ORed between the halt and break interrupt lines to force a restart trap.

The microprocessor reads the AI0 to AI7 input lines and arbitrates the interrupt priority according to Table 3-1. In addition, the state of AI1 to AI5 is reproduced on TDAL 12–8 lines during the acknowledge cycle. TDAL 11–8 lines are used to decode the interrupt acknowledge. TDAL 12 reflects the state of the VEC L signal: asserted VECT L for external interrupts; unasserted for internal interrupts.

The interrupt logic (Figure 3-1, Sheet 1) receives interrupt requests from interface devices and applies them to the microprocessor. The microprocessor will acknowledge the highest priority interrupt if its priority is higher than the current microprocessor status word priority. There are 16 interrupts available, and either one or all can be active at the same time. Four interrupts become active for low going signals, and the remaining 11 become active for high going signals. Three of these inputs are ORed to produce one level 6 request, and two other interrupts are ORed to produce one latched level 7 request. Five of the

Table 3-1 Designated Interrupts

			Coded Input Vector					
Interrupt	Priority		AI-1	AI-2	AI-3	AI-4	AI-5	
Source	Signal	Level	TDAL8	TDAL9	TDAL10	TDAL11	TDAL12	
Halt/ break	HLTRQ/ BREAK	Non-maskable	X	X	X	X	X	Restart address
DMA request	DMARQ	Non-maskable	X	X	X	X	X	DMA
Power-Fail/ QB reset	PFAIL/ BINIT	Non-maskable	X	X	X	X	X	24
Real-Time clock	RTCRQ	6	0	1	0	0	1	100
Counter timer	RQ104	6	0	1	0	1	1	104
Two-Port RAM	RQ120	5	1	0	0	0	1	120
Two-Port RAM	RQ124	5	1	0	0	1	1	124
Two-Port RAM	134	5	1	0	1	1	1	134
QIR	RQ130	5	1	0	1	0	1	130
SLU1 receive	DLRRQ	4	1	1	0	0	1	60
SLU1 transmit	DLXRQ	4	1	1	0	1	1	64
SLU2	UPDRQ	4	1	1	1	0	1	70

remaining 10 interrupts are latched and stay active until reset. The 12 enabled interrupts are clocked through internal flip-flops by BCAS L asserting during the present microprocessor transaction. These outputs are used to produce an encoded vector. The vectors are enabled by the PI H input of the present transaction.

The interrupt codes and their priority levels are listed in Table 3-1. When the PI H output is asserted, the microprocessor looks at the interrupt inputs and initiates an interrupt acknowledge (IAK) transaction for any interrupts with correct priority following completion of the current microprocessor instruction. The microprocessor coded input is placed on TDAL bus bits 8–12. Each TDAL bit is listed with its proper AI interrupt code input in Table 3-1. These five TDAL bus bits are inputs to the acknowledge decoder that is enabled when the microprocessor starts an IAK transaction which occurs after an IAK input becomes active. These inputs are decoded to determine which interrupt was acknowledged and outputs a low to negate the interrupt request. SLU1 transmitter and receiver interrupt lines are reset by wire ORing and asserting the acknowledge decoder output low. TDAL 12 represents the state of the VEC L input when the microprocessor acknowledges the interrupt: a low for external interrupts and a high for internal interrupts. The signal is used to determine if an external vector interrupt acknowledge must be initiated.

3.10.2 Ready Logic

The ready logic (Figure 3-20) provides the microprocessor READY input and is used to control the microprocessor cycle slip function. The microprocessor cycle slips when the READY input is being pulsed from high to low, while RAS L is asserted; the cycle slip function will be inhibited when READY H input is set high. The pulsed READY H input causes the T-11 to slip one, four, or more cycles. Both cycle function control lines, CSLIP1 L and CSLIP4 L, are outputs of the memory address decoder. These signals are initiated by either the T-11 or DTC chips with the DC339 encoding the control signals. For any cycle slips to occur, the following five lines with proper states must be present.

1. DPRDY L is unasserted (high), indicating the two-port RAM is not being accessed.
2. EIAK L is unasserted (high), indicating no external interrupt acknowledge is being serviced.
3. RQTM L is unasserted (high), indicating no timeout request is being serviced.
4. SEL0 H and SEL1 H are unasserted (low), indicating no direct memory grant was given.
5. RAS L is asserted (low), indicating a read operation.

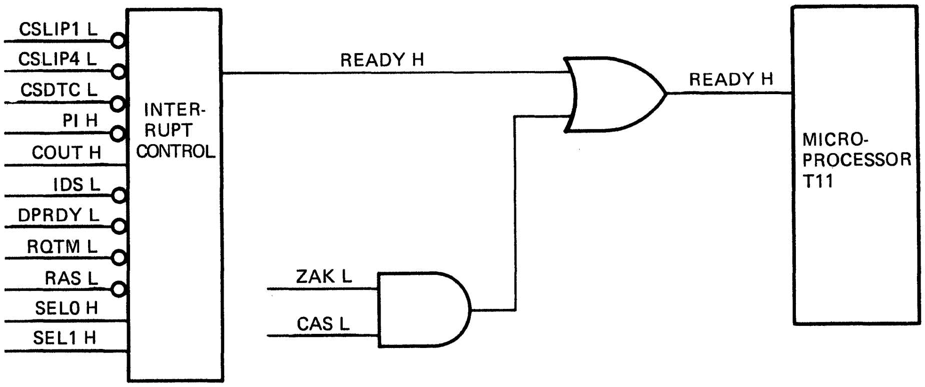


Figure 3-20 Ready Logic

During normal operations one T-11 cycle slip occurs for the T-11 interface with SLU1, SLU2, or the counter/timer chip. When PI L is asserted and clock COUT goes from low to high, the READY line goes from high to low. This high-to-low transition of the READY line is ORed. The other OR gate input ensures gating of the READY line during any DTC interrupt acknowledge or at the beginning of a CAS L strobe. This is due to the DTC operating at double the T-11 speed. The high-to-low gate outpuı transition causes the T-11 to slip one cycle.

Before the next low-to-high transition of COUT H, PI H becomes unasserted, causing the READY line to go back to a high.

For four cycle slips to occur, the DTC chip select (CSDTC L) and cycle slip (CSLIP1 L) lines must be asserted, while CSLIP4 L remains unasserted. During assertion of data strobe (IDS L) and a low-to-high transition of the clock line (COUT H) into the DC339, the READY line begins to pulse. After the fourth READY transition the READY line remains high, inhibiting any more T-11 cycle slips to occur.

Multiple cycle slips occur when the two-port RAM (DPRDY L) line remains unasserted with cycle slip (CSLIP1 L) and (CSLIP4 L) lines both asserted. When this happens, each low-to-high transition of the clock (COUT) line into the DC339 causes the READY line to pulse low. This low causes the T-11 to cycle slip once. READY line pulsing stops when the two-port ready line becomes asserted (DPRDY L).

3.10.3 Restart Interrupts

The restart interrupts (Figure 3-21) are defined as BREAK H and RHALT H going to the microprocessoı AI7 input. RHALT H is a bus arbiter command and BREAK H is a console command. The restart interrupts are developed in gate arrays DC339 and DC340. DC340 outputs RHALT H as BRQ140 H and is applied as an input to DC339.

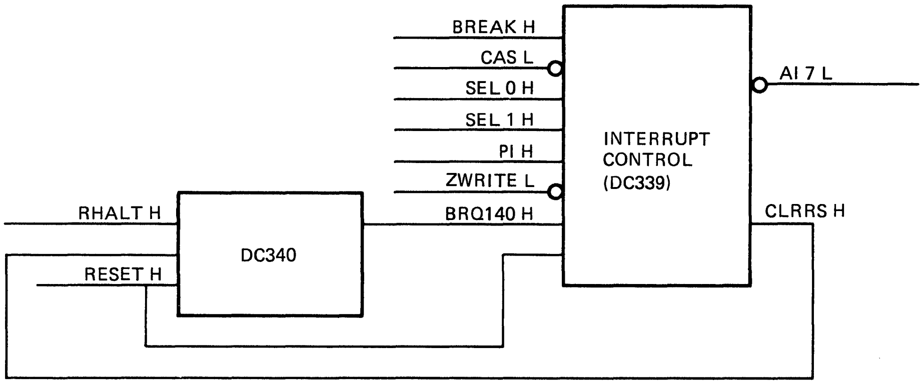


Figure 3-21 Restart Interrupts

During a reset, the logic associated with both restart interrupts are reset. The restart interrupt output line AI7 L is unasserted high.

During normal operations, BREAK H and RHALT H remain unasserted. If either line becomes asserted due to a restart interrupt, the interrupt is clocked out on line AI7 L when CAS L is asserted. The interrupt is read into the T-11 at the leading edge of PI H, causing a nonmaskable level 7 interrupt. During this interval the CLRRS H line is unasserted. CLRRS H becomes asserted after the interrupt has been read into the T-11. To ensure that an interrupt was not cleared before it was read into the T-11, four lines –

SEL1 H, SEL1 L, ZWRITE, and PI H – control the CLRRS H line. The CLRRS H line is asserted after SEL0 H and SEL1 H are unasserted, indicating the T-11 is in control of the bus (no direct memory grant); ZWRITE L is unasserted, indicating that a T-11 read has been done; and PI H is asserted indicating an interrupt read has been done by the T-11. When all these conditions are met, CLRRS H asserts. This line clears the restart interrupt logic and resets the AI7 L line to the unasserted state.

3.10.4 Power-Fail (PFAIL)

The PFAIL H output is connected to microprocessor AI6 input and is recognized as a power-fail nonmaskable interrupt (Figure 3-22). When acknowledged, the microprocessor traps through octal address 24 to access the PC and the PSW for a user's power-fail routine. It is suggested that user power-fail routines should set the PSW to 340 to prevent other interrupt requests from disrupting the power-fail routine. During a reset instruction the power-fail logic is reset.

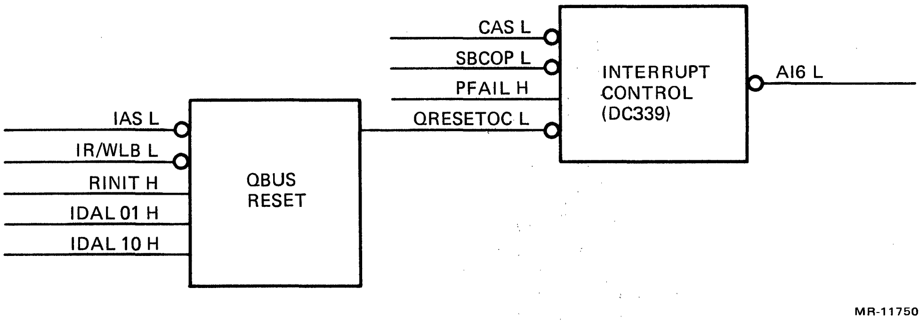


Figure 3-22 Power-Fail Interrupt

The input lines for the AI6 L interrupt are SBCOP L, PFAIL H, and RINIT H. The inputs are used differently depending on which mode the board is operated in. In the standalone mode, SBCOP L asserted when the SBC ID switch is placed in the standalone position. This low inhibits the PFAIL H line from initiating a Q-Bus power-fail interrupt. In the peripheral processor operating mode, the PFAIL H asserts a minimum of 9 ms before loss of Q-Bus dc power. This signal allows an orderly KXT11-CA power-down sequence thereby minimizing erroneous microprocessor cycles. The RINIT H line enables the SBC to detect Q-Bus resets. When a Q-Bus reset occurs with the reset detect function enabled, QRESETOC L is asserted. For additional details about QRESETOC L, see Paragraph 3.9. In addition, RINIT H can still detect Q-Bus resets and cause AI6 L interrupts even though the board is terminated to a Q-Bus backplane but operating in the standalone mode. If the board is not terminated to a Q-Bus backplane, the RINIT H line can be externally wired and controlled. When externally controlled, the bus initialize bit in control status register D must be enabled. See Chapter 5 for the bit enable information. If the bit is not enabled, the RINIT L line is inhibited. When both RINIT H and PFAIL H are inhibited, no AI6 L interrupts are detectable.

In the Q-Bus mode after a power-up or reset instruction is performed, all power-fail interrupt input lines become unasserted. A power-fail interrupt is generated upon detection of either PFAIL H or RINIT H assertion. When either line is asserted, CAS L latches AI6 L interrupt low. The AI6 L low is read into the T-11 during assertion of PI H.

3.10.5 External Vector Cycle

Two devices can generate an external T-11 interrupt vector. They are the DTC and PI/O counter timer chips. An external vector cycle starts when one of these devices generates an interrupt that is driven onto interrupt lines AI1 L–AI5 L. During the next microprocessor read cycle, this interrupt request data is strobed, when PI H is asserted, into the microprocessor. This assertion indicates to the microprocessor that an externally generated interrupt vector address is to be read in. The microprocessor now performs an external IACK transaction. The interrupt request address (AI1 L–AI5 L) is output on TDAL 8–12 when SEL1 H is asserted. The microprocessor asserts RAS L, enabling the interrupting device to drive the external vector onto TDAL bus lines. The external vector is read into the microprocessor when RAS L goes unasserted. During an external IACK cycle, one microprocessor cycle slip occurs, and ZAK L is asserted, due to the relative faster speed of the external interrupting devices.

3.10.6 Internal Vector Cycle

An internal vector cycle occurs when the interrupt control logic receives an internal interrupt request. The request is encoded and driven onto interrupt request data lines AI1 L–AI5 L. Request data line AI5 L (VECT L) is unasserted for internal interrupt vectors. The interrupt request data is strobed into the microprocessor during PI H assertion. The microprocessor reads the encoded interrupt request data lines (TDAL) and points internally to the requested internal interrupt vector address. The microprocessor performs an IACK transaction by driving encoded request data onto TDAL lines 8–12 enabling the interrupting device to be reset.

3.10.7 DMA Request/Grant

The DMA request is connected to the AI0 input to the microprocessor. The DMA request is received by the microprocessor during any read, write, fetch, or ASPI transaction. A request is not acknowledged by an IAK transaction, but is acknowledged by the microprocessor, asserting both SEL0 and SEL1 outputs to initiate a DMA grant. (See Paragraph 3.15 for a discussion of DTC operations.)

3.10.8 End Of Pass (EOP)

The end of pass (EOP L) signal, when asserted, stops all DMA operations which are in progress. (See Figure 3-23.) The EOP logic is reset when SEL0 H and SEL1 H are asserted and monitors several SBC functions centering around DMA operations using the Q-Bus. An EOP L assertion occurs for the following.

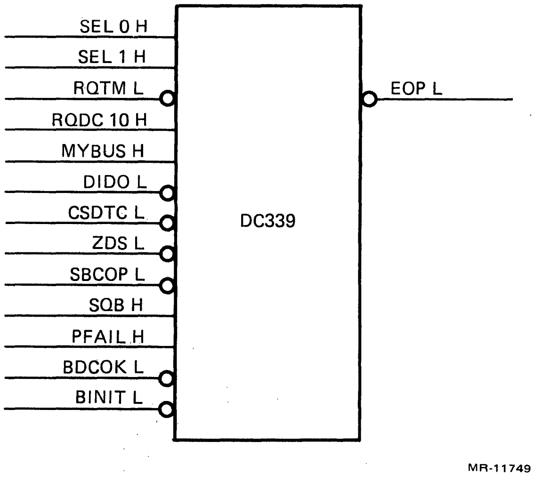


Figure 3-23 End of Pass (EOP) Logic

1. The DTC tries to address a local nonexistent memory location (RQTM L asserts).
2. A DTC request for the Q-Bus is not granted in less than 10 us. Request for the bus RQDC10 H is asserted, ownership of the bus has not been granted, MYBUS L is unasserted, and both data strobes DIDO L (output) are unasserted.
3. DTC has local bus control and tries to address its own internal registers. This condition exists when CSDTC L and ZDS L occur together.
4. When in SBC mode (SBCOP L asserted) and a Q-Bus address is referenced (SQB H asserts).
5. When a Q-Bus address is referenced during a Q-Bus power-failure (PFAIL H asserts), Q-Bus power-down sequence (BDCOK L unasserts), or a Q-Bus initialization (BINIT L asserts).
6. DTC references to a Q-Bus address and no response from the Q-Bus is received in less than 10 μs. The data in/data out signal DIDO L asserts for a Q-Bus read/write cycle and starts the 10 μs count clock.

3.11 ADDRESS LATCH

The address latching logic (Figure 3-1, Sheet 1) has 16 transparent latches. The output is always enabled. New incoming TDAL data becomes address bus lines ADB01–15 during the leading edge assertion of ddress strobe IAS L. The address bus goes to the logic sequencer and memory address decode logic. The address bus is common to on-board memories, remaining stable during assertion of address strobe IAS L.

3.12 MEMORY DECODE

The memory decode logic has two field programmable logic arrays (FPLA) that decode applied address bits and three memory map bits. The memory map bits can be jumpered providing eight memory maps, shown in Chapter 6, for the user. The FPLA is enabled when the disable decode line DISDE H is unasserted. Signal DISDE H is asserted, and the FPLA output is inhibited during Q Bus operations. The address lines ADB01–ADB15 come from the address latch.

3.13 RAM MEMORY

The static RAM memory (Figure 3-24) is a 16 K × 16-bit memory that has two 8 K × 8 high byte chips and two 8 K × 8 low byte chips. The memory is selected by assertion of CSRAM1 L and CSRAM2 L inputs going to pin E1. The memory is addressed by bits ADB01–ADB13, and 16-bit data is read from or written to via TDAL bits 00–15. The IR/WLB L assertion selects read/write operations for a low byte, and IR/WHB L assertion selects read/write operations for a high byte. The write inputs go to the OE input pins.

The user can supply the static RAM memory with +5 V battery backup. The user must supply +5 V to SBC pin AV1 and install the battery backup jumper between two jumper posts. See Chapter 6 for detailed information.

3.14 USER MEMORY SOCKET SITES

The RAM/ROM memory (Figure 3-25) provides the user with two 28-pin sockets to accept either 24-pin or 28-pin standard +5 V chips. The sockets can hold either EPROMs, PROMs, RAMs or +5 V only EEPROMs. There is a high byte and low byte socket. Both sockets use the CSKTA L output from the memory address decode as chip enables. (See Chapter 6 for memory maps.) The IR/WLB L, IR/WHB L, and the IREAD L signals from the DC339 are used to produce a high byte read/write and a low byte chip read/write. There are 15 jumper posts available for memory configurations.

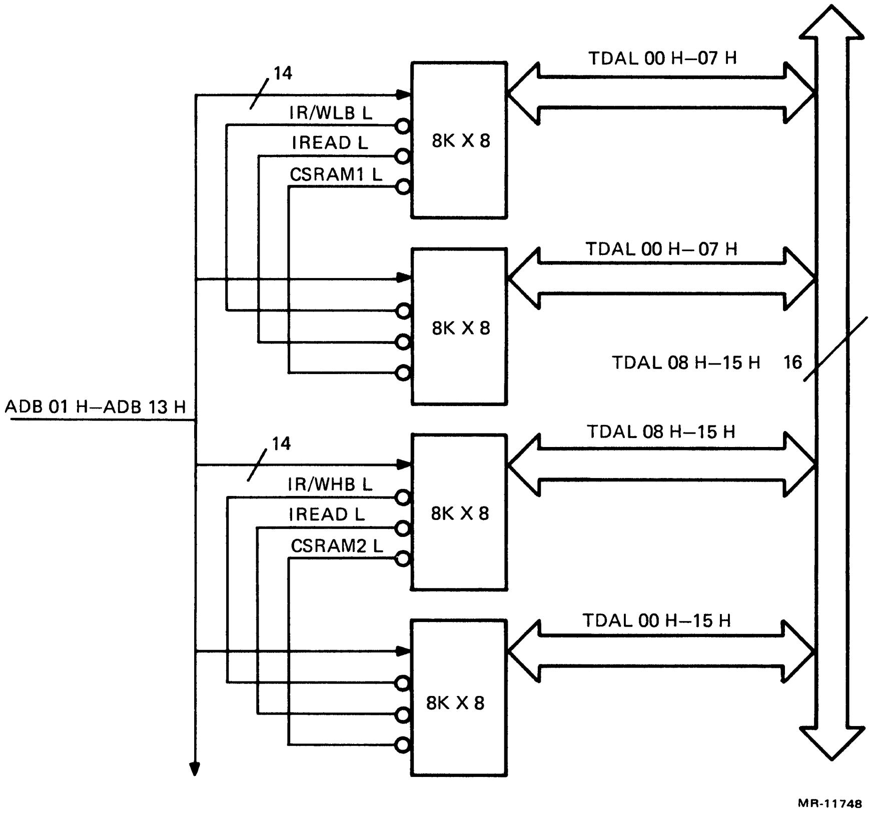


Figure 3-24 RAM Logic

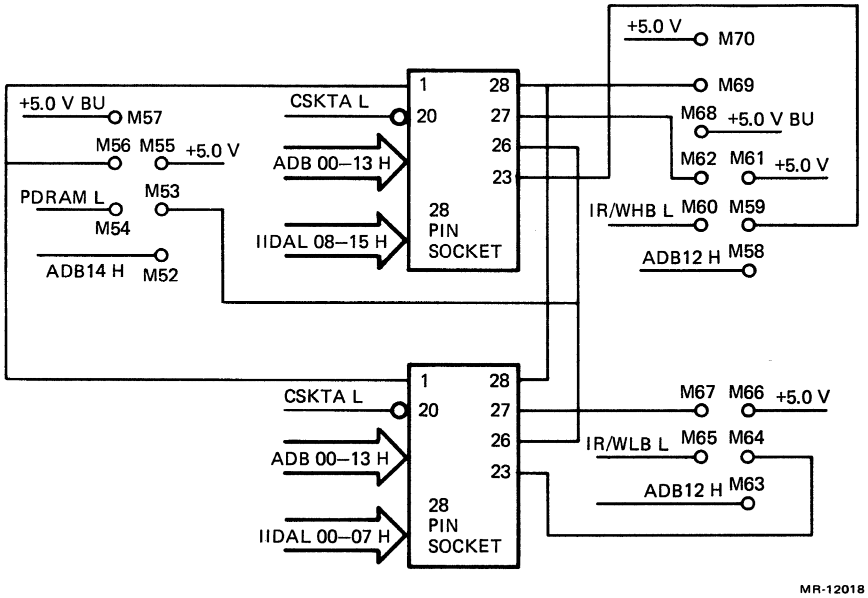


Figure 3-25 User Socket Sites

3.15 NATIVE FIRMWARE SOCKET SITES

The native firmware sockets sites (Figure 3-26) can accommodate 24-pin 4 K × 8, 28-pin 8 K × 8 PROMs or additional 24- or 28-pin RAMs with battery backup. Native firmware PROMS control the power-up selftest, device initialization, program booting, program loading, system primitives, and diagnostics. Local address decode logic allows only 3 Kw of the 4 Kw portion of this PROM above 160000 (octal) to be directly mapped into the memory space at any one time. So, to use the last 1 Kw portion, the lower 1 Kw portion can be overlaid with the last 1 Kw portion of PROM. KXTCSR A, the local status control register, controls the overlay bit. See Chapter 4 for detailed information.

Both sockets use the CSKTB L output from the memory address decode for chip select. In addition, IREAD L signal from the DC339 is used to perform reads of the PROM.

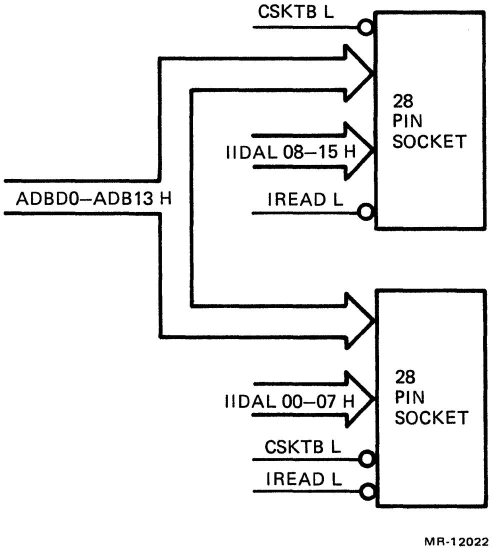


Figure 3-26 Native Firmware Socket Sites

3.16 DTC OPERATIONS

The direct transfer controller (DTC) is a 16-bit DMA controller addressable by the local microprocessor as an I/O device. See Figure 3-27 for DTC control signal lines. The DTC supports two channels and performs data transfers, using DMA operations, between any local 16-bit address and any 16-, 18- or 22-bit Q-Bus address. Word, high byte, and low byte data transfers are supported locally. Only word data transfers are supported across the Q-Bus.

The DTC has the ability to load its control parameters from memory in a chaining sequence. During a chaining sequence the only operation the microprocessor must perform is to load a control parameter table into the DTC channel and issue a start chain command. The DTC can operate independent of the microprocessor except for an interrupt at the termination of the operation. In addition to transferring data, the DTC can compare data from the source device to the content of a pattern and mask register and then notify the microprocessor when a match or no match occurs.

DMA operations can be interleaved between the T-11 and both DTC channels, or they may occur in various burst sizes. Channel A or B can be used to service hardware requests from SLU2 channel A (transmitter and receiver) or from the PI/O timer counter when transferring data via the serial port (J1) and the parallel port (J4).

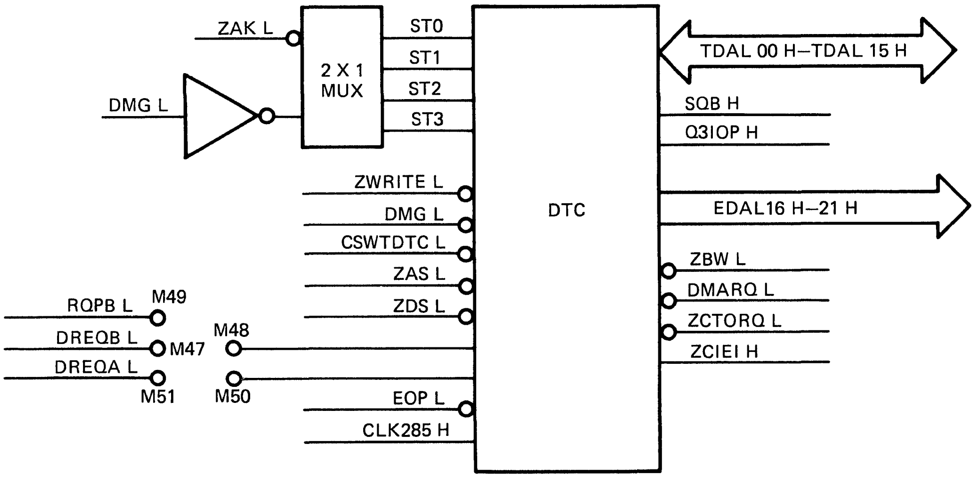


Figure 3-27 DTC Control Signals

3.16.1 Flowthrough Transactions

Data transfers are performed in a two-step sequence called flowthrough transactions, where a source is read and a destination is written. The data is held in a temporary register between the source and destination portions of the transfers. The DTC can perform three types of transfers: local, Q-Bus, and I/O.

3.16.1.1 Local Transfers – Local data transfers occur between two on-board memory locations. (See Figure 3-28.) Local transfers are initiated by the microprocessor when the DTC chip select line CSWTDTC L is asserted, and the read/write line, ZWRITE L, is selected, unasserted for a read or asserted for a write. Since the DTC is neither doing a vectored acknowledge (ZAK L is unasserted) nor has it received a direct memory grant yet, DMG L is unasserted, forcing the output of the 4 × 2 multiplexer to all lows. This indicates an internal DTC operation is being done. In addition, the DTC is currently a local bus slave, forcing both ZAS L and ZDS L to be DTC inputs. When the address on the TDAL bus is valid, the microprocessor asserts the address strobe ZAS L. The byte/word output is active (ZBW L) indicating that a word, high byte, or low byte operation was selected. Next, the data to be transferred is driven onto the TDAL bus and when valid, the microprocessor asserts the data strobe ZDS L line. This strobes the data into the DTC. After the data is stored in the temporary DTC registers, the DTC asserts DMARQ L to request local bus mastership.

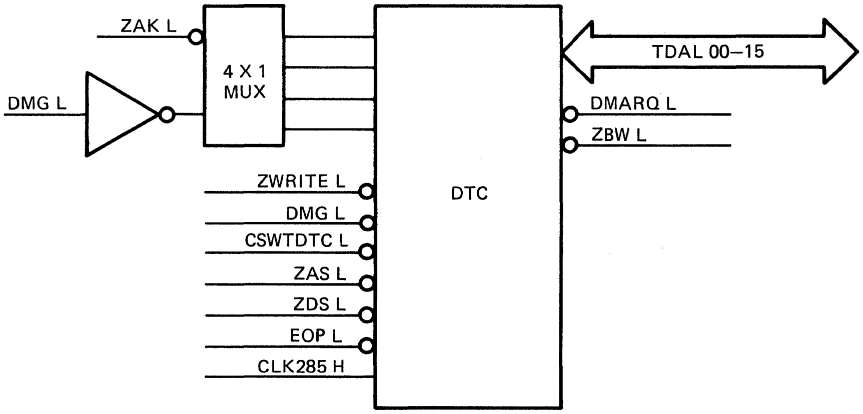


Figure 3-28 Local DMA Control Signals

The microprocessor responds via the DC339 gate array by asserting the memory grant line DMG L. The DTC now is master of the local bus. Control lines ZAS L, ZDS L and ZWRITE L become outputs to the local bus. First, ZAS L asserts when the address is valid, then ZDS L asserts, indicating valid data is present on the bus. Finally, ZWRITE L asserts if a write to the bus is being done. At the end of the transfer, the DC339 logic asserts the end of pass EOP L line. This input into the DTC terminates, causing DMARQ L to go unasserted and forcing the microprocessor to remove the grant (DMG L). The DTC is once again a local bus slave and the microprocessor is local bus master.

3.16.1.2 Q-Bus Transfers – Q-Bus DTC transfers are similiar to local transfers in that the DTC must acquire mastership of the local bus exactly the same way. See Figure 3-29 for the Q-Bus transfer control signal lines. After acquiring the local bus, the DTC must request and then acquire mastership of the Q-Bus via DC010 operations. To initiate the DC010 operations, SQB H asserts. Once the DC010 acquires bus mastership, the DTC performs the requested transfer via the SBC's XDAL bus transceivers (see Figure 3-1, Sheet 2). If the Q-Bus I/O device is addressed, the device address is located in the top 4 K of memory. To address these devices the DTC asserts QIOP H line. When addressing 18- or 22-bit Q-Bus addresses, the EDAL 16–22 lines are used. At the completion of the transfer, lines SQB H and DMARQ L unassert. Unasserted SQB H terminates DC010 operations and returns bus control back to the bus arbiter.

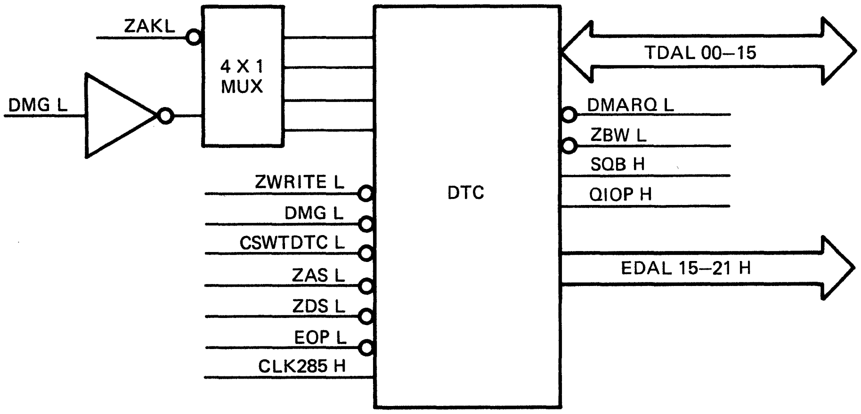


Figure 3-29 Q-Bus DMA Control Signals

3.16.1.3 I/O Transfers – Both channel A and B of the DTC support hardware DMA requests. See Figure 3-27 for DTC control signal lines. The hardware requests can be from three sources: PI/O counter timer and SLU2 channel A for both receiver and transmitter. The DTC hardware requests from SLU2 are DREQA L for the transmitter and DREQB L for the receiver. In addition, the DTC hardware request from the counter timer request line is RQPB L. Requests DREQA L and RQPB L can by jumpered to channel A of the DTC. All three hardware requests are maskable. See Chapter 5 for details on masking.

3.16.2 Exceptions

Flowthrough transactions are operated on by the DTC until a terminal count is read or an end of pass (EOP L) signal is received. This low-going signal terminates the activity of the active channel. If neither channel is active, the EOP line is ignored. The signals used to develop EOP are breifly described below. The EOP signal is developed in the DC339 gate array logic. For more details pertaining to EOP signal, see Paragraph 3.8.1.7.

3.16.2.1 Timeout – When the local timeout request line RQTM L asserts, the EOP L line asserts. This returns control of the local bus to the microprocessor.

3.16.2.2 SBC Mode – When operating in the SBC mode, anytime the DTC tries to read or write to Q-Bus address an EOP L assertion will occur.

3.16.2.3 Power-Fail – When a Q-Bus power failure occurs, the EOP L line asserts. This enables the microprocessor to regain control of the local bus so an orderly power-down cycle occurs.

3.16.2.4 Bus Initiation – When the DTC is in control of the local bus and a Q-Bus initiation occurs, the DTC responds by returning control of the local bus to the microprocessor. The microprocessor controls the local bus for the duration of the bus initiation.

3.16.2.5 Bus Halt – When a Q-Bus halt occurs and the DTC is in control of the local bus, the DTC operations do not stop. The RHALT L line assertion is read into the microprocessor's interrupt line, causing the microprocessor to trap to the restart address.

3.17 SERIAL I/O UNIT

The KXT11-CA SBC has three serial line units (SLUs) which operate via three I/O ports. SLU1 operates over a 10-pin I/O port, J3. SLU2 consists of two channels: channel A operates over a 40-pin I/O port, J1, and channel B operates over a 10-pin I/O port, J2.

3.17.1 Serial Line Unit 1 (SLU1)

SLU1 is driven, as shown in Figure 3-30, by the DC319 DLART (DL compatible asynchronous receiver/transmitter) chip. The DC319 chip has a programmable output baud/second rate, either software or hardware (jumper) selectable. See Chapter 5 for the DC319 software programmable register control information; Chapter 6 contains hardware baud/second rate jumper selections. The hardware selectable baud rates are from 300 to 38.4 K baud/second in eight steps. If neither hardware or software baud rates are selected, the DC319 chip powers up to a default baud rate of 300.

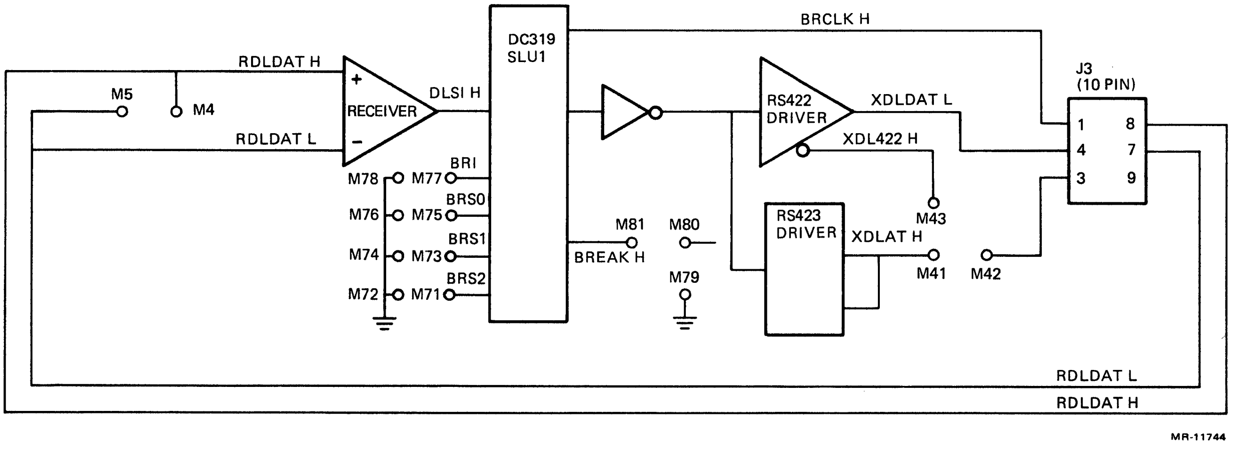


Figure 3-30 SLU1 Transmit/Receive Configuration

Outgoing serial data transmitted by the DC319 (DLSO) is inverted and is driven by two transmitter drivers. One driver provides compatibility for differential (EIA RS422) transmission and the second driver provides compatibility for single-ended (EIA RS423) transmission. In addition, an external real-time clock output from DC319 (BRCLK) is provided for transmission. The real-time clock output rate is 76.8 KHz.

Serial data coming in on the DC319 can be supported for either single-ended or differential input conditioning. The incoming serial data is received on J3 pins 7 and 8. A single jumper is installed for differential receiver operation and no jumper is installed for single-ended operation. See Chapter 6 for jumper details.

3.17.2 Serial Line Unit 2 (SLU2)

SLU2 is driven by a two-channel multiprotocol serial controller (UPD 7201). Local data transfers to/from the controller and TDAL bus are in parallel form; I/O port data transfers between the controller and drivers are in serial form. These transfer protocols are for both transmit and receive operations. Simultaneous operation of both channel A (see Figure 3-31) and B (see Figure 3-32) is possible, each capable of supporting asynchronous or synchronous communications.

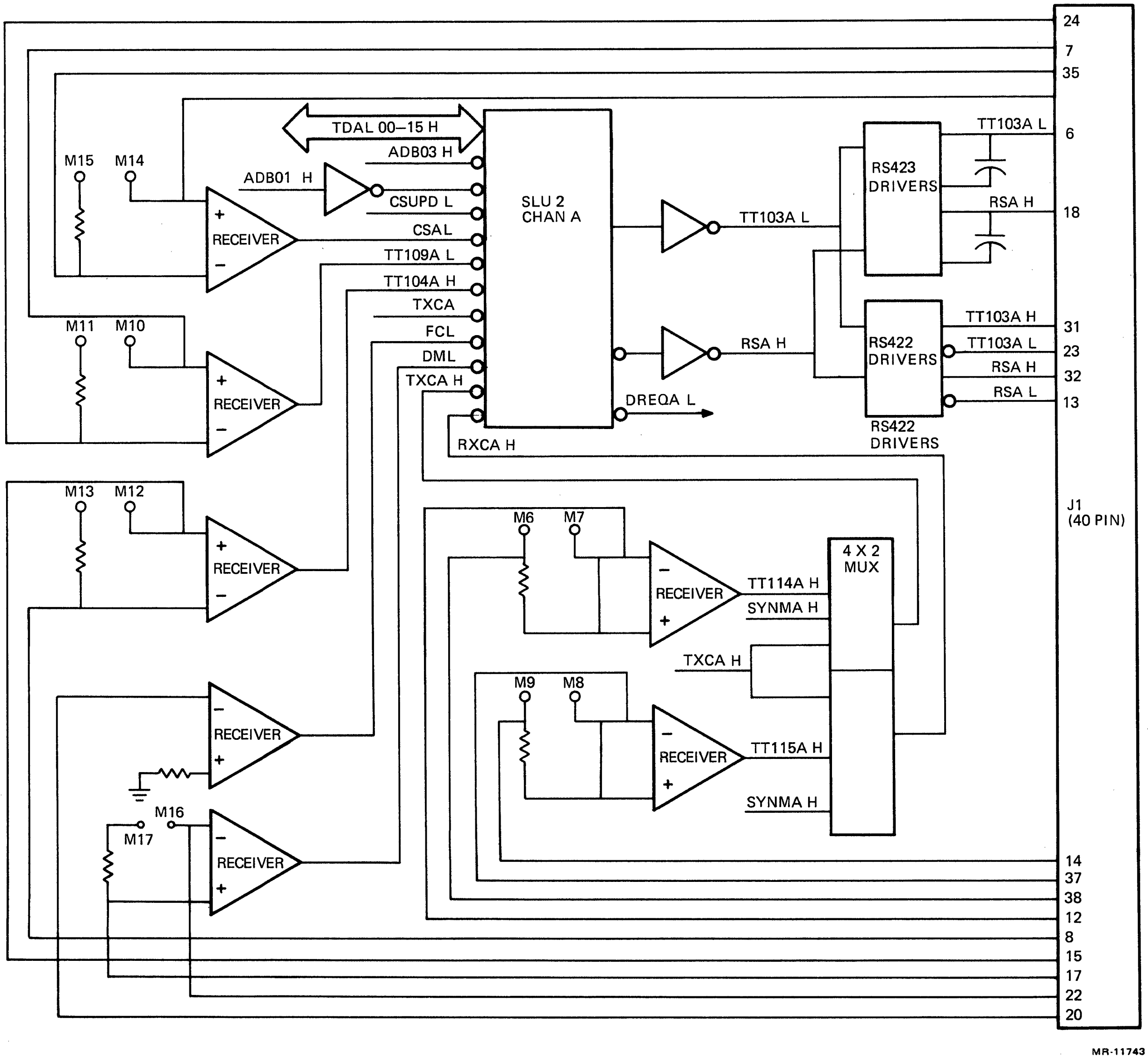
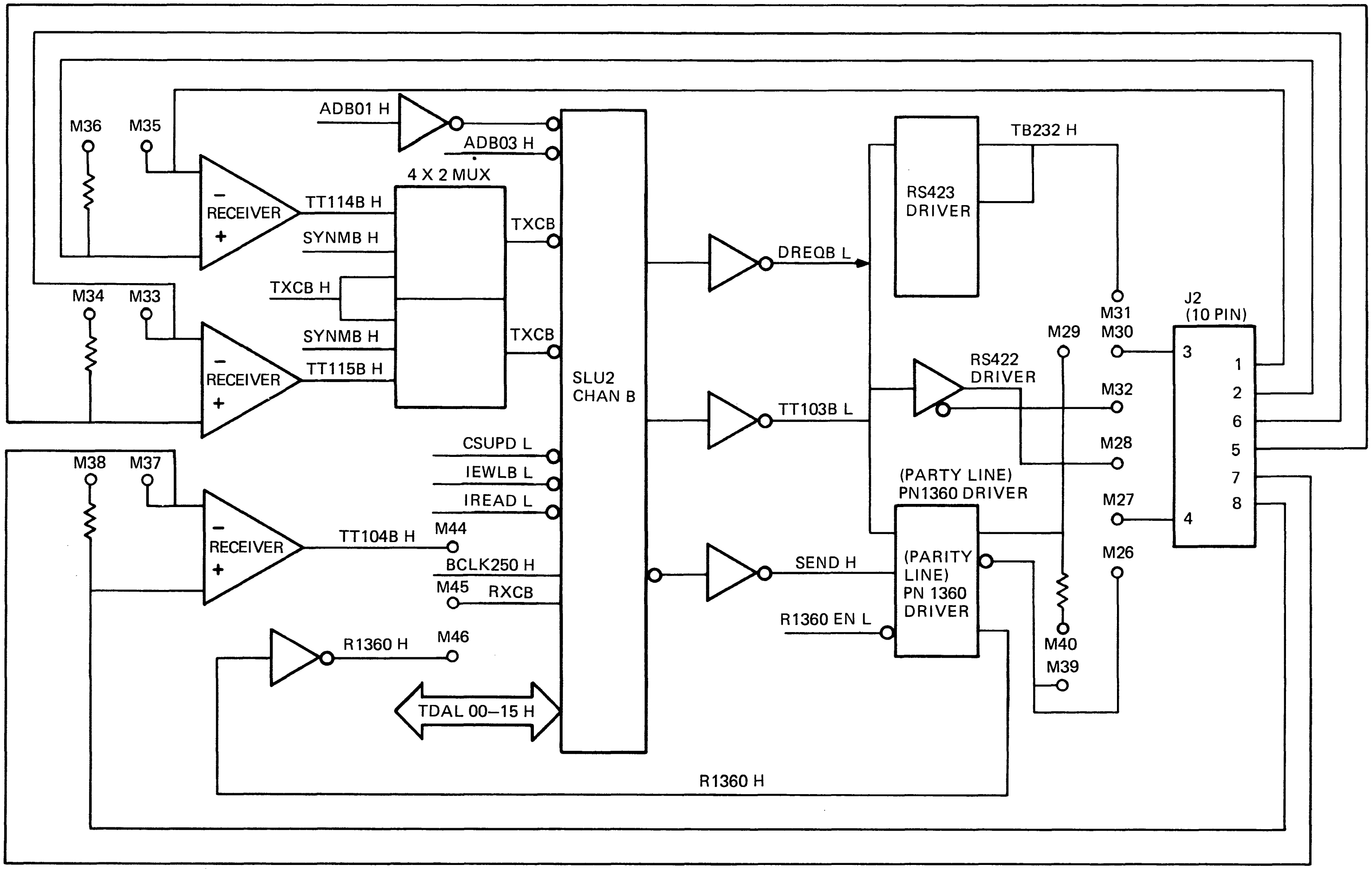


Figure 3-31 SLU2 Channel A Receiver



MR-11742

Figure 3-32 SLU2 Channel B Transmitter/Receiver

Chip select line CSUPD L must be asserted for all controller operations. The controller is driven by a 7-MHz clock signal (BCLK250 H). During SBC power-up the INIT L signal is asserted. This disables both channel A and B outputs and clears all the internal controller registers. The native firmware writes into these registers at the end of the power-up cycle. After this is complete, signal ADB01 H selects either data or control transfers – a high for control and a low for data. Signal ADB03 H is used to select either channel A or channel B operations. An asserted ADB03 H selects channel B and an unasserted ADB03 H selects channel A. Assertion of the IREAD L signal transfers data out onto the local TDAL bus, and assertion of IEWLB L transfers control parameters into the controller. See Chapter 6 for more details on configuring channels A and B.

3.17.2.1 Channel A – The transmit inputs consists of eight parallel TDAL bus lines (TDAL 00–08 H), a clear to send CSA L, and a transmit clock rate. The TDAL bus lines are used to transfer data to the controller in parallel form. CSA L can be used as a transmitter enable sent by the receiver at the opposite end of the communication link. The transmit clock rate is used to clock outgoing transmitter data. The transmit clock rate is software programmable for either local or remote clock via a 4 × 2 multiplexer. One multiplexer input is the programmable timer output (TXCA H) and the other is the received clock rate (TT114A H) output. The multiplexer output is controlled by the software programmable line, SYNMA H. A high selects TT114A H and a low selects the programmable timer output, TXCA, for data transmissions.

The output serial data (TT103A L) is inverted and applied to both RS423 and RS422 driver inputs. Both driver outputs are wired to I/O connector J1. The request to send RSA H is also inverted and is applied to both RS423 and RS422 driver inputs. The outputs are wired to connector J1. The user must wire the J1 cable for either RS422 or RS423 transmission, depending on the application. The DMA request line, DREQA L, is used as DMA interrupt input to the local DTC.

The channel A receiver consists of a serial receiver input and a receiver clock signal. The incoming serial data (TT104A H) can be jumpered for either RS422 or RS423 reception through the receiver driver. The receiver clock rate is software programmable for either a local or remote receiver clock via a 4 × 2 multiplexer. One input to the multiplexer is a programmable timer output, TXCA, and the other input is a received clock, TT115A H. The multiplexer output is selected by the SYNMA H signal. A high into the multiplexer selects TT115A H and a low selects the programmable timer output, TXCA, for data reception. The receiver output is driven onto the TDAL bus in parallel form.

3.17.2.2 Channel B – The transmitter inputs consists of 8 parallel bus (TDAL 00–08 H) data lines, an incoming call (IC L), and a receiver clock rate. The TDAL bus lines are used to transfer data to the controller in parallel form. The incoming call, IC L, is used as a channel B receiver enable signal. The receiver clock is software programmable for either local or remote receiver clock. Receiver clock rate, TXCB, is selected via a 4 × 2 multiplexer. One input of the multiplexer is a programmable timer output, TXCA H, and the other input is the receiver clock rate output, TT115B L. The multiplexer output is selected by SYNMA H. A high selects TT115B L and a low selects TXCB as the clock.

The transmitter outputs are serial data (TT103B L) and party line enable (SEND H). The transmitter output is inverted and is applied to all three output drivers: RS422, RS423, and party line. The user must jumper in the proper transmitter driver for the application. If the party line is jumpered in, enable signal SEND H must be asserted. The serial data is transmitted via I/O connector J2.

The channel B receiver input signals are serial data, data mode carrier detect (DM L), and receiver clock rate. The user must jumper the receiver input line for either RS422, RS423, or party line receiving. Carrier detect (DM L) can be used as a receiver enable sent by the receiver at the opposite end of the communication link. The receiver clock is software programmable for either a local or remote receiver clock. The local receiver clock is the programmable timer output, TXCB, and the remote clock is the transmitter output, TT115B H, from the opposite end of the communication link. The local and remote clocks are fed to a 4 × 2 multiplexer. The multiplexer output is selected by signal SYNMA H. A high selects TT115B H and a low selects TXCB for data transmissions. The receiver output is driven onto the TDAL bus in parallel form.

3.18 PARALLEL I/O UNIT

The parallel I/O port is implemented with the 8036 counter/timer (ZCIO) chip and connects to the user's interface through I/O connector J4. (See Figure 3-33.) The I/O buffers are capable of providing high impedance inputs, with 600 mV of hysteresis and either passive internal pull-up driver outputs or TTL-compatible driver outputs. The ZCIO chip configures port A buffers, in nibbles, for either input or output operation. In addition, each bit within a nibble can be configured for bidirectional operation. Port B is configurable as an 8-bit port (two 4-bit nibbles) or as a bit port with the direction of each bit controlled by the ZCIO chip. The four port C bidirectional lines can be individually configured as input or output bits controlled by the ZCIO chip. See Chapter 5 for configuration details on the software programmable ZCIO registers.

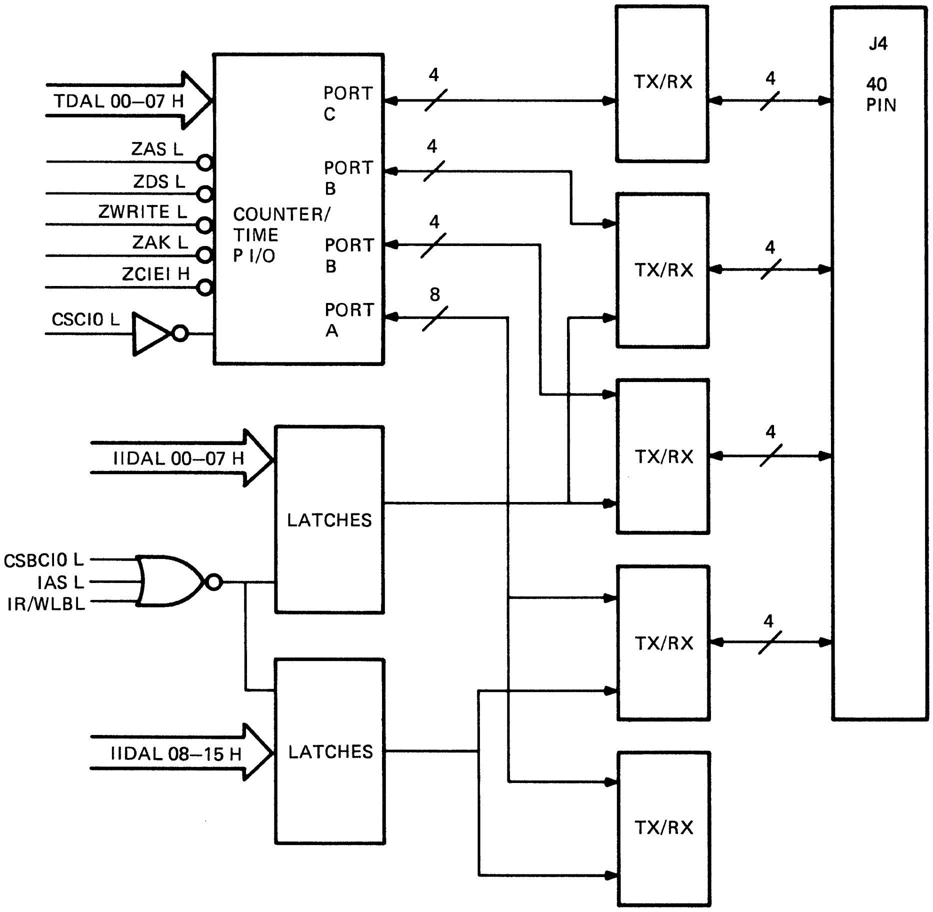


Figure 3-33 Parallel I/O (PIO) Port Configuration

3.19 POWER-UP

The power-up logic (Figure 3-34) senses the application of +5 V power to the SBC and initiates a power-up sequence. When the +5 V is first applied, C1 charges up. Before C1 is charged, PDRAM L is unasserted. This low is inverted twice through two inverters and becomes TPUP L. Asserted TPUP L inhibits the NAND gate output, clears the power-fail flip-flop, and resets the RS flip-flop. The output of the AND gate, PUP H, asserts, causing the microprocessor RESET L output signal to assert. The reset flip-flop Q output, LDCLO H, remains unasserted. The +5 V charges C1 through R5 until the threshold level of the inverter is reached. This occurs at approximately 2.6 Vdc and 50 ms after +5 V was applied. TPUP L unasserts high, enabling the unasserted GPFAIL H signal to preset the PUP/power-fail flip-flop to a high on the Q output. This high combines with the unasserted PUP L to gate the NAND gate. The NAND gate output, PUP H, goes low, enabling microprocessor operations. The SBC has successfully completed power-up. The GPFAIL H signal prevents PUP H from negating until BPOK H asserts.

During normal peripheral processor operation, Q-Bus power signals, QDCLO L and GPFAIL H remain unasserted. If a Q-Bus power-down sequence occurs, the first signal to become asserted is GPFAIL H. After a minimum of 4 ms, the QDCLO L signal becomes asserted low. This low sets the power-fail to a high. The high is inverted to a low, enabling C1 to discharge through R28. This low (discharge) PD RAM L is inverted twice, supplying a low TPUP L line. T PUP L clears the PUP flip-flop to a low, resets the power-fail flip-flop, then goes high. The microprocessor is forced to initiate a power-up sequence.

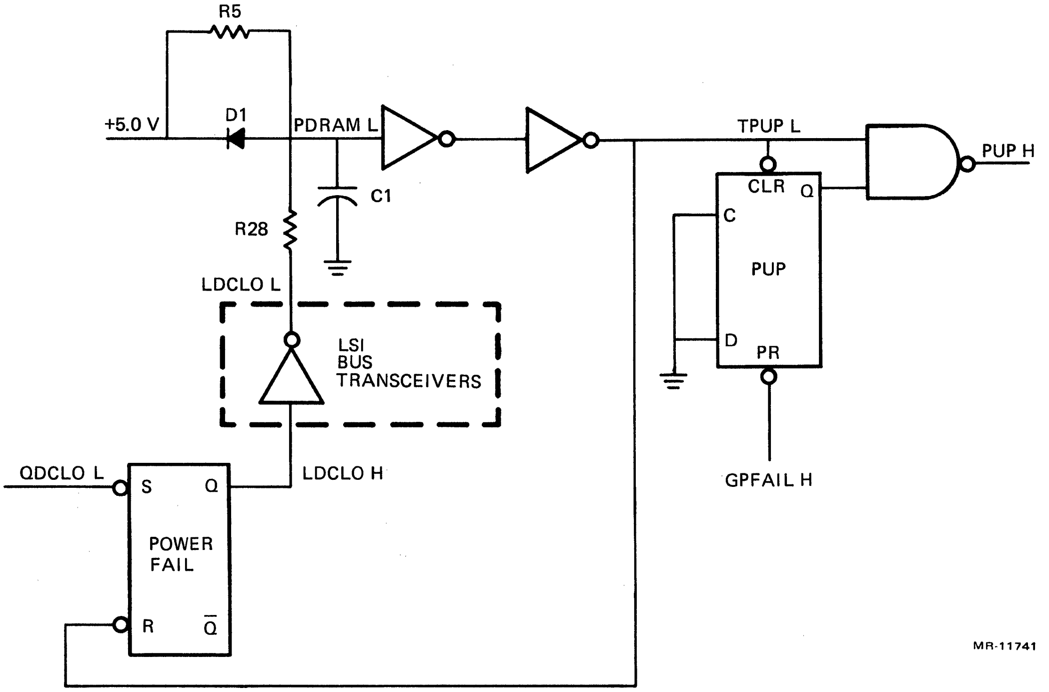


Figure 3-34 Power-Up (PUP) Logic

CHAPTER 4
NATIVE FIRMWARE

4.1 INTRODUCTION

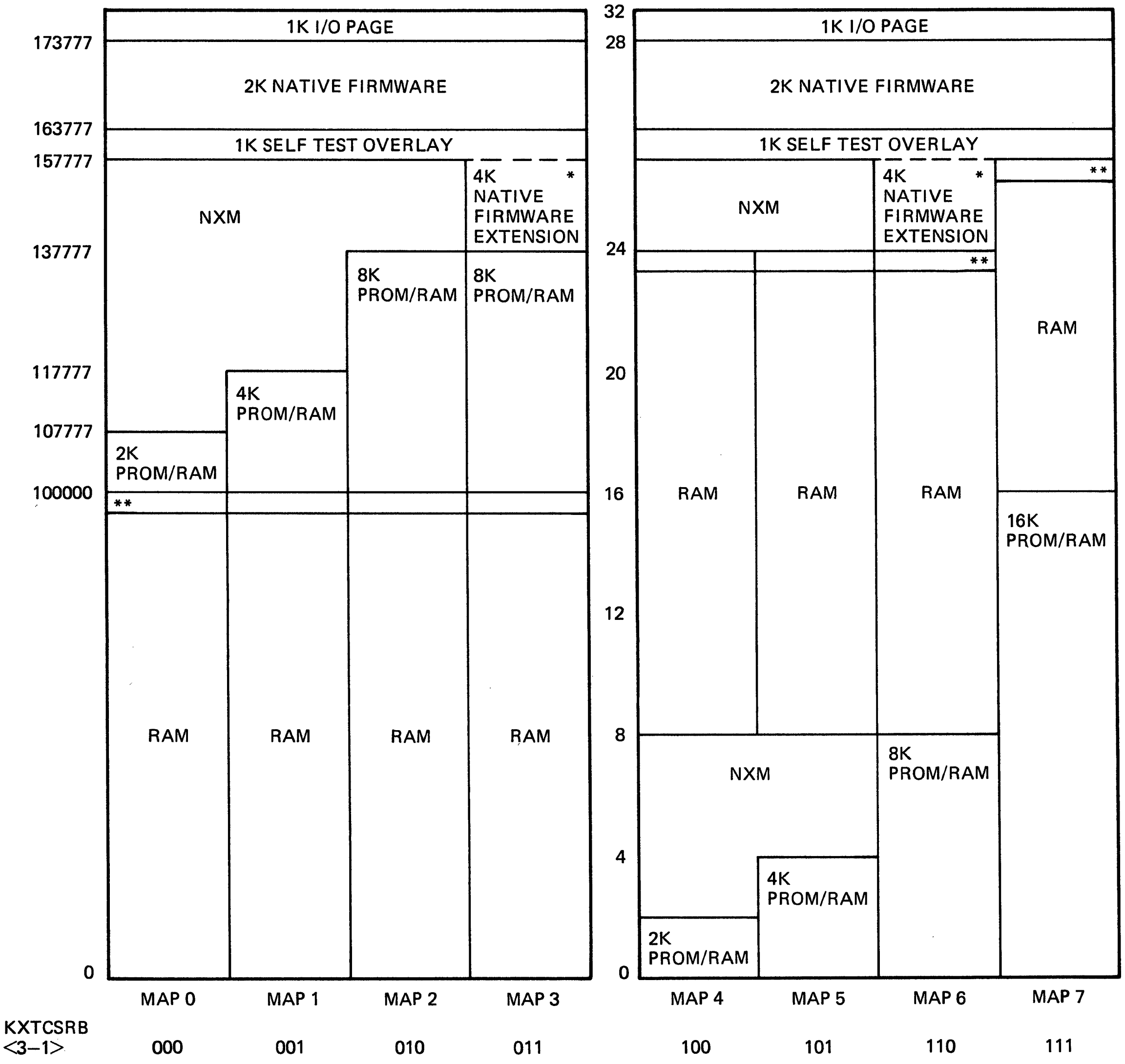
The native firmware ROM programs are routines executed by the SBC on power-up (address 173000), on receiving a console break or on receiving Q-Bus commands. ROM programs execute the selftests, initialize Q-Bus interfaces, and perform processes that indirectly allow an application program to be loaded and executed. In addition, ROM programs service various local events, NXM (nonexistent memory), HALT (local), and B HALT (Q-BUS HALT). The KXT11-CA native firmware ROM listing is located in a separate document. See Chapter 1 for document ordering details.

4.2 NATIVE FIRMWARE

The top 64 bytes of the native RAM are reserved for native firmware usage. See Figure 4-1 for the native firmware location within any of the eight possible memory configurations. The user should not attempt to address this portion of RAM, with the exception of the top two words which are the battery backup flags. Any user data present in this space is altered by the native firmware. The firmware sets the stack pointer to the RAM location directly below this reserved space before control is passed to the application code (default user SP). Because this portion of the native RAM is reserved, the user sockets should be mapped low if they contain RAM.

The native firmware interfaces with the following.

1. Boot/Selftest switch (power-up)
2. Two-Port RAM command register
3. SLU1 console serial ODT
4. LED display
5. Memory map selection



*NOT CURRENTLY SUPPORTED

**THE TOP 64 BYTES OF NATIVE RAM ARE RESERVED BY NATIVE FIRMWARE

MR-11673

Figure 4-1 Memory Configuration Showing Native Firmware

4.3 BOOTSTRAP/SELFTEST SEQUENCE

The bootstrap sequence brings the SBC from an initial power-up or reinitialized state to running user application programs. Several native firmware routines are used. There are two entry points into the bootstrap sequence, power-up and reinitialization. Figure 4-2 shows the power-up sequence.

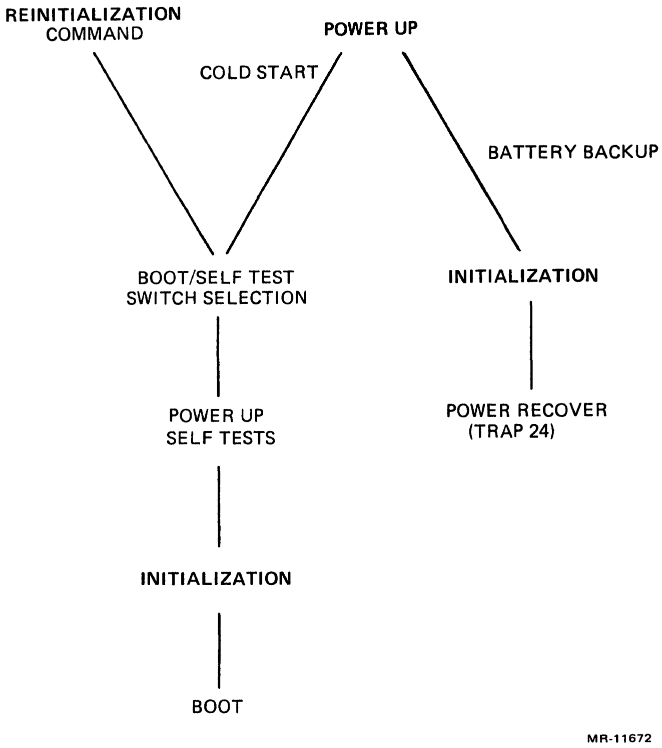


Figure 4-2 KXT11-CA SBC Power-Up Sequence

4.3.1 SBC Set-Up Routine

After receiving a power-up or reinitialization command, the native firmware initializes the SBC hardware control registers. The set-up routine disables all maskable interrupts and causes the two-port RAM (TPR) to appear as all zeroes to any Q-Bus reads.

4.3.2 Initialization

The native firmware initializes the local I/O portion of the SBC. The following local devices are initialized.

1. The direct transfer controller (DTC) chip (is first reset)
2. The serial driver for SLU2 channels A and B
3. The parallel interface device (PIO).

See Chapter 5 for more details on specific device register contents during reset and initialization.

4.3.3 Battery Backup

The native firmware must determine if the native RAM is provided with battery backup. The test for battery backup consists of comparing the contents of two RAM locations for specific values. The values are 125252 (octal) in the top word and 52525 in the word below native firmware. If provided with battery backup, the microprocessor will trap to the power-up address (24). For details on native firmware detection of battery backup, see the status register description in Chapter 5.

4.3.4 Selftests

The native firmware determines which, if any, selftests are to be executed. The boot/selftest switch is used to select the mode of operation. A portion of the selftests are called the auto selftests. They are a subset of the total selftest functions available. The auto selftests include the following.

1. CPU test
2. RAM test, native and user (if RAM in user sockets)
3. ROM test, native and user (if selected)
4. Control status registers (CSR), nonexistent memory test of all native CSRs and TPR (read only test)
5. DMA test, test of local DMA only transfers

If the RAM or native ROM tests fail, the SBC will not allow the application code to run and places the SBC in a dead state, displaying a fatal error. The same happens if the TPR portion of the CSR test fails when the SBC is in the peripheral processor mode. The reporting of any test failures is done via the LEDs and the TPR. The test results are placed in TPR registers 2 and 3 at the completion of the auto selftests. Register 3 is overwritten only if an error was encountered and contains the valid discrete error code for the last test run that found an error.

4.3.5 Booting

Upon successfully passing the selftest routine, the boot/selftest switch is used to determine which booting routine is to be used. The boot routine selects either booting application code from a ROM or a TU58 drive unit. If booting from a ROM is selected, the native firmware transfers control to that ROM code by emulating a trap to location 24. When a TU58 drive unit is selected, booting is done via SLU1 console I/O port J1. Once the bootstrap begins receiving boot blocks from the TU58, receiving boot blocks from the TU58, a comparison of the first byte to octal value of 240 to 277 is made. If a match occurs, then the boot block is considered valid and all remaining blocks are transferred. If no match occurs, the bootstrap continues to retry alternating between units 0 and 1.

4.4 COMMAND REGISTER

The (first) register 0 (local address 175000) in the two-port RAM (TPR) is called the command register (Figure 4-3). Any time this register is written to by a Q-Bus processor, a nonmaskable restart interrupt (173004) to the SBC is generated. This interrupt is serviced by the native firmware which interprets the command sent.

There are three types of commands that can be sent from the Q-Bus arbiter to the command register: control, selftest and ODT. For details on the bit definitions of the command register see Chapter 5.

The procedure for issuing any type of command to this register is as follows.

1. Test the status register for valid state.
2. Test the command register for zero.
3. If the command register is all zeros, the SBC is not available.
4. Write the command into the command register.
5. Wait for the command register to be cleared.
6. Check the status register for a command error.

Q - BUS ADDRESS	LOCAL ADDRESS
BASE + ID + 36	175036 R/W
34	175034 R/W
32	175032 R/W
30	175030 R/W SIGNAL LOCAL CPU ON WRITE (LEVEL 5 VECTOR 134)
26	175026 R/W
24	175024 R/W
22	175022 RO STATUS, WRITES TIMEOUT
20	175020 R/W SIGNAL LOCAL CPU ON WRITE (LEVEL 5 VECTOR 124)
16	175016 R/W
14	175014 R/W
12	175012 RO STATUS, WRITES TIMEOUT
10	175010 R/W SIGNAL LOCAL CPU ON WRITE (LEVEL 5 VECTOR L 120)
06	175006 R/W
04	175004 R/W
02	175002 RO STATUS, WRITES TIMEOUT
BASE + ID + 00	175000 R/W SIGNAL LOCAL CPU ON WRITE (NON - MASKABLE RESTART TRAP)

MR-12169

Figure 4-3 Two-Port RAM (TPR)

4.4.1 Control Commands

There are six control commands, described in the following paragraphs, that can be issued to the command register by a Q-Bus device. The native firmware detects these commands and responds with a specific function.

4.4.1.1 Trap Command (Bit 0 Set) – The trap command causes the firmware to emulate a trap routine. The trap address must be placed in TPR register 2 using the PSW from TPR register 3. The native firmware will trap to the specified address using the PSW. Just before the trap is started, the command register (0) will be cleared acknowledging that the command is being done. This command is used in the debug stage for testing trap handling routines.

4.4.1.2 DMA Load Command (Bit 1 Set) – This command causes the native firmware to perform a chain load via the DMA controller (DTC). TPR registers 2 and 3 are used to pass data parameters that are loaded into the chain control registers for channel 1 of the DTC. Register 2 contains the segment tag parameter, and register 3 contains the offset tag parameter. After the data parameters are loaded into the chain control registers, the native firmware starts the chain load. When the DTC has either completed the transfer or aborted, the command register is cleared. If the transfer is aborted, an error bit in TPR register 1 is set, and the contents of the DMA status for channel 1 is placed in TPR register 3 before the command register is cleared. In either case of completion, the native firmware enters a wait state.

4.4.1.3 Reinitialize Command (Bit 2 Set) – This command is equivalent to a power-up. The native firmware performs a power-up without battery backup. With this command, the boot/selftest switch can be overridden. TPR register 3 is used to pass the selftest/boot test selection codes. See Chapter 6 for the selftest codes.

4.4.1.4 Q-Bus ODT Command (Bit 3 Set) – This command causes the native firmware to enter the Q-Bus ODT mode. In this mode the internal microprocessor registers are saved, the contents of TPR register 1 are saved, and the command register accepts the Q-Bus ODT commands. On completion, TPR register 1 state field is set to the Q-Bus ODT mode and the command register is cleared.

4.4.1.5 Show Command (Bit 4 Set) – This command causes the SBC to present its boot/selftest switch setting and memory map switch settings in TPR register 3. Register 3 can then be viewed from a Q-Bus device. See Chapter 5 for register 3 bit information. On completion of the Show command, the SBC enters a wait state.

4.4.1.6 No Operation Command (Bit 5 Set) – This command is reserved for Digital Equipment Corporation use only. Issuing this command when running application code on the SBC may cause mishandling of the restart interrupt.

4.4.2 Selftest Commands

There are 11 test commands that can be issued by writing to the TPR command register from the Q-Bus. A brief description of each test follows. (For more register bit pattern information, see Chapter 5.) If a parameter is associated with a test command that parameter is issued prior to issuing the command. Parameters are issued via TPR3. On completion of any test, the results are placed in TPR registers 2 and 3 and the command register is cleared. Register 2 contains cumulative error information and register 3 contains discrete error information. Execution of any test may alter the state of the SBC control status registers (CSRs) and RAM. All tests involve the use of interrupts, where applicable, only if the native RAM is mapped to low memory location 0 or the user socket contains RAM. To map the RAM to location 0, set the memory map to a position that maps the RAM low or set the boot/selftest switch to position 10. If RAM is not set low, the interrupt portions of tests are not performed. On completion of any test, the SBC enters the wait state.

4.4.2.1 CSR Test Command (Bit 0 Set) – This is one of the initial power-up or reinitialization tests. When this command is received, the native firmware attempts to read the local command/status registers (KXTCSR A, B, C, and D).

4.4.2.2 RAM Test Command (Bit 1 Set) – This test command enables a nondestructive test of the SBC RAM to be performed. The test is also run on power-up or reinitialization if enabled. If RAM is present in the user sockets, it, too, is tested. The native firmware determines if RAM is present in these sockets by attempting to alter the first socket address location. If the contents at this address can be successfully altered and restored, then the RAM test is performed.

4.4.2.3 ROM Test Command (Bit 2 Set) – This test performs a checksum of the native ROM contents (and the user ROM, if selected). TPR register 3 is set to one before this command is issued if the user ROM is to be tested, or set to zero if it is not. The algorithm for the ROM test is listed in Chapter 5.

4.4.2.4 CPU Test Command (Bit 3 Set) – This test command causes the microprocessor to exercise a subset of the instruction set. The result is presented in TPR register 3.

4.4.2.5 BEVENT Test Command (Bit 4 Set) – This test verifies that the line clock interrupt can be enabled, generated, and disabled. The generated interrupt is at priority level 6 through internal vector 100. For this test to be performed, RAM must be mapped to zero.

4.4.2.6 SLU1 Test Command (Bit 5 Set) – This test causes the console serial line unit to be tested. The loopback connector must be in place for this test to be performed. RAM must be mapped to zero for the interrupt portion of this test to be performed.

4.4.2.7 SLU2 Test Command (Bit 6 Set) – This test performs the multiprotocol interface chip test. Both SLU2 channel A and B loopback connectors must be in place for channel A and B to pass the tests. The channel tested is selectable via the TPR. RAM must be mapped to zero for the interrupt portion of this test to be performed. The channels should be tested individually to provide meaningful error data.

4.4.2.8 PIO Test Command (Bit 7 Set) – This command causes a test of the Z8036 chip. The test checks the parallel I/O port and the timers. The loopback connector must be used in performing this test. RAM must be mapped to zero for the interrupt portion of this test to be performed.

4.4.2.9 DMA Controller Test Command (Bit 8 Set) – This test command causes the DMA controller to be tested. The test is done by performing a DMA transfer from local memory to local RAM, from local memory to Q-Bus RAM, from Q-Bus RAM to Q-Bus RAM, and from Q-Bus RAM to local RAM. Before this command can be issued, TPR register 3 must be loaded with the high 16 bits of the base address of the 4 K word of Q-Bus RAM to be used.

4.4.2.10 QIR Test Command (Bit 9 Set) – This command causes the QIR (Q-Bus interrupt register) interface to be tested. Q-Bus interrupt vector address (0–1776 octal) must be placed in TPR register 3 before the test command can be issued. When the Q arbiter receives the interrupt, it signals the SBC that an interrupt was received by performing a reset instruction.

4.4.2.11 Two-Port RAM (TPR) Test Command (Bit 10 Set) – This command initiates a read and write test of both ports of the TPR. The test uses the DMA controller in testing the TPR, so the DMA controller test (Paragraph 4.3.2.9) should be done prior to testing the TPR.

4.4.3 Q-Bus ODT Commands

Once the SBC is in the Q-Bus ODT mode, by receiving the Enter Q-Bus ODT command (Chapter 5), the command register format is changed, but procedures for issuing ODT commands remain the same. This also includes how the SBC responds to a command error. The Q-Bus processor-initiated ODT command details follow.

4.4.3.1 Open and Examine Memory Command Code – The memory specified by TPR register 3 is opened and its contents are presented in TPR register 2.

4.4.3.2 Open and Examine Register Command Code – The microprocessor general-purpose register specified by TPR register 3 is opened and the contents are presented in TPR register 2. See Paragraph 4.5 on opening registers.

4.4.3.3 Deposit Command Code – This command initiates the deposit of the data presented in TPR register 2 into the open memory location or register.

4.4.3.4 Go Command Code – This command instructs the native firmware to restore all CPU registers to the user values and to transfer control to the location specified by TPR register 3. The PSW is set to zero and a reset instruction is performed.

4.4.3.5 Proceed Command Code – This command is the same as the Go command except it uses the current saved PC value, restores the PSW, and does not perform a reset.

4.4.3.6 Exit Mode Command Code – This command exits the SBC from the Q-Bus mode and waits for a control command. When completed, the previously running application code context is lost. The command register format returns to the control/test format and the SBC waits for the next command.

4.5 SERIAL ODT

In addition to boot programs, the SBC native firmware contains code for performing the serial ODT functions via the console I/O port (J1). This is necessary because the T-11 microprocessor does not implement ODT in microcode. The following ODT commands are implemented in macrocode. The ODT monitor responds to the following ASCII character encoded commands.

1. "n/" – Open memory location or register (e.g., 1000/ or R1/)

2. "/" – Reopen last memory location or register previously closed

NOTE

> **1. If the "/" command is used without a target and a target was never specified in the current ODT session, a default address of zero is used.**
>
> **2. If the "/" command is used after an error, the command is not accepted and an error is issued.**

3. CARRIAGE RETURN – Close memory/register

4. LINE FEED – Close last memory/register and open the next

5. "^" – Close last memory location or register and open preceeding

6. "R" or "r" – Specify register to be opened (e.g., R1), (set of: R0 through R7, RS)

   - "0"..."7" – CPU register R0 through R7 (e.g., R1)
   - "S" or "s" – CPU status register (e.g., RS)

7. "," – Specify range to be examined (e.g., 1000,1100/ or R0,4/)

8. "G" – Restore all CPU registers to user values. Perform a reset instruction.

9. "P" – Same as "G" command, except a reset instruction is not performed.

NOTE

> **1. The last location is opened on completion of the command.**
>
> **2. Number following "," must be greater than the preceding number. If not true, an error prompt is displayed.**

The user enters serial ODT mode by sending a BREAK signal to the console serial interface on the SBC.

NOTE
The BREAK signal is received if the BREAK jumper is installed.

When the SBC is in the serial ODT mode, the TPR is disabled and all commands from the Q-Bus are ignored. Serial ODT mode can also be entered if the boot/selftest switch is in position 4 on power-up or reinitialization. BREAK commands entered while the SBC is in the serial ODT mode cause the present command to be aborted and the original BREAK PC to reappear with a new prompt. This can be used to abort a range examine, as shown in the following example.

```
entry      -   (BREAK)

response   -   123456
               @

entry      -   0,100/

response   -   @0,100/011565
               000002/177776
               000004/005015
                 .
                 .
                 .

entry      -   (BREAK) [ In the middle of the range examine]

response   -   123456
               @
```

4.5.1 Autobaud

Serial ODT autobauds if SLU1 (console I/O port) is configured for software baud rate operation and the baud rate has not been previously programmed. The autobaud routine determines the proper operating baud rate by comparing the character received to the value of a RETURN character. The procedure for causing the ODT to perform this function is to enter several RETURNs after the SBC is in the ODT mode until the following ODT response is printed (PC value, CR, LF, @). The autobaud routine works for a terminal set at 300, 600, 1200, 2400, 4800 or 9600 baud.

NOTE
The baud rate can be configured for either hardware or software baud rate operation. If the IOP is configured for hardware operation (see Paragraph 6.2.9), autobaud is inhibited.

4.6 LED DISPLAY

The native firmware sets the state of all four LED display indicators using four bits in KXTCSR C. The results of the SBC selftests and the state of the firmware are displayed by the four LEDs. The LEDs indicate one of two types of errors: first encountered nonfatal error or a fatal error. A table of the error displays and a descripiton of each error is shown in Chapter 6. It is suggested that the user should not alter the LED status.

4.7 MEMORY MAP CONFIGURATION

The native firmware reads the state of three bits in KXTCSR B. The three bits are user selectable for eight possible memory maps. For details on the eight memory maps available, see Chapter 6.

4.8 RESTART HANDLING

The KXT11-CA firmware must interpret the cause when the T-11 microprocessor is vectored to the restart address 173004. Five events can cause the SBC to interrupt to the restart address. The native firmware tests for each of these events in the following order and services the first one to occur. For detailed register bit and flag information, see Chapter 5.

1. Break – The native firmware determines if the Break function was the cause of the restart by testing the low three bits of the boot/selftest switch. If BREAK caused the interrupt, all three bits will be ones. These three bits are located in KXTCSR D.

   The native firmware clears any break interrupts by reading the RBUF of SLU1. It also clears the NXM and TPR interrupt indicators. The flag bits of the KXTCSR D are used to determine what action to take. Then these bits are normally cleared. If cleared, the native firmware enters the serial ODT mode.

2. TPR Interrupt – This event is indicated by bits in local KXTCSR D (control status register). The native firmware responds to the event by clearing the TPR interrupt indicator and the NXM indicator, then decoding the command received for the Q-Bus and taking the appropriate action.

3. Nonexistent memory (NXM) – This event is determined through another indicator by a bit in KXTCSR D. If this indicator is set the native firmware clears it. The firmware then attempts to emulate a trap through the vector at 4. This process requires a valid stack. If the stack is determined to be invalid (pointing to a nonexistent memory location), the native firmware sets the SP to the default SP value (top of native RAM – 100), sets the stack error flag of the status register (TPR word 1), and emulates a trap to 4. If there was no stack problem during the trap emulation, the stack error flag of the status register is cleared.

4. Halt instruction – When the T-11 microprocessor executes a halt instruction, the peripheral processor traps to the restart address (173004). The native firmware determines if this is the cause of the restart by checking the location pointed to by the return PC on the stack minus 2. If the location contains a 0, the cause of the restart is assumed to be a halt instruction. If bit 4 of the status register, TPR word 1, is set to a "1" the native firmware enters serial ODT. If the bit has not been set to a "1" (cleared on initialization), the native firmware emulates a trap through the vector at 10.

   NOTE

   This bit should be set when using MAC DEBUG or POS DEBUG.

5. BHALT – Receiving a Q-Bus halt (BHALT) with the BHALT enable bit set in KXTCSR D causes an interrupt to the restart address. The native firmware tests the BHALT bit in KXTCSR D register to determine if this is the cause of the interrupt. If this is the case, the native firmware emulates a trap through the vector at location 24, leaving the BHALT bit in KXTCSR D register set. This is the last test for the cause of the restart. If the cause is not found, the native firmware performs a return from interrupt.

4.9 BOOT EXIT

When the native firmware transfers control of the SBC to the application software, the SBC assumes the following state.

- All TPR registers are cleared except for TPR registers 1, 2, and 3.
- All I/O interrupts are disabled.
- Parallel port A interrupt vector is set to 200.
- Parallel port B interrupt vector is set to 204.
- Parallel port timer interrupt vector is set to 210.
- DTC channel A interrupt vector is set to 110.
- DTC channel B interrupt vector is set to 114.
- DTC internal registers are at indeterminate values.
- The LEDs are set to non-native code running state or error display.
- The stack pointer is set to first RAM address below the reserved RAM space.

CHAPTER 5
PROGRAMMING INFORMATION

5.1 INTRODUCTION

This chapter details all the user-addressable registers for the KXT11-CA single-board computer. This chapter also describes the usage of each software programmable register bit. Appendix E is a list of each addressable register in the KXT11-CA local environment.

5.2 CONTROL STATUS REGISTERS (KXTCSR)

The local processor has four I/O page addressable registers available for controlling and monitoring the SBC system environment. KXTCSR A is an 8-bit read/write byte- or word- addressable register at address 177520. KXTCSR B is an 8-bit byte- or word-addressable read-only status register at address 177522. KXTCSR C is an 8-bit byte- or word- addressable read/write register at address 177524. These three low-byte registers are available through the 8255A-5 chip. During power-up the native firmware initializes the three registers by writing a 212 at address 177526. This causes all low-byte register outputs to be driven to a zero state. Rewriting the mode control word reinitializes the 8255 chip. The high byte of these registers must be treated as erroneous. The 8255A-5 chip is not altered by the reset instruction. KXTCSR D is a 16-bit word-addressable control and status register at address 177530 and is cleared by a local reset.

5.2.1 KXTCSR A Register

Figure 5-1 shows the bit positions and the bit definitions of the KXTCSR A register at address 177520. Table 5-1 defines the KXTCSR A register.

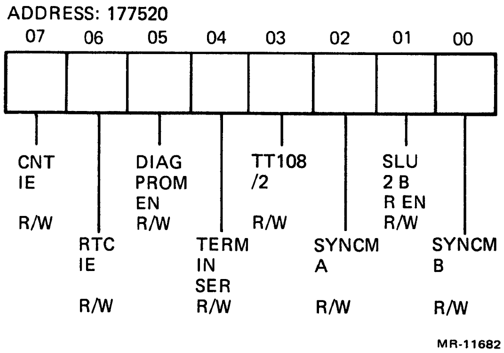


Figure 5-1 KXTCSR A Register

Table 5-1 KXTCSR A Bit Description

Bit	Name	Status	Function
07	CNT IE	Read/write	Programmable counter interrupt enable, cleared on power-up, or by writing a zero. When cleared the programmable counter/timer 2 interrupt is inhibited from being asserted. When set by writing a one, the counter/timer can interrupt the T-11 on level 6 vector 104 each time the OUT pin is driven high.
06	RTC IE	Read/write	This bit is cleared on power-up or by writing a zero. When cleared, the RTC interrupt is inhibited from being asserted to the local processor. When set by writing a one, the RTC interrupt is enabled.
05	DIAG PROM EN	Read/write	This bit is cleared on power-up or by writing a zero. It is set by writing a one, which causes a 1 K word portion of the extended selftest PROM to be visible at location 163777–160000. When cleared, the other 1 K word portion of the PROM is visible at the same addresses. This, in conjunction with the directly addressable 2 K word portion mapped to 173777–174000, allows a total of 4 K word of dedicated firmware for ODT, selftest, diagnostics, device initialization, and boots to reside in the I/O page.
04	TERM IN SER	Read/write	This bit is cleared on power-up or by writing a zero. When cleared, the modem is signaled that the IOP is not available for service or that the modem should appear busy so that incoming calls will not be connected. When written with a one, the IOP is in service and incoming calls can be connected.
03	TT108/2	Read/write	This bit is cleared on power-up or by writing a zero. when cleared, the modem is signaled to be removed from the communication channel following completion of the transfer of all data presented before this signal was turned off. When set by writing a one, the modem prepares to be connected to the communication channel and maintains the connection.

Table 5-1 KXTCSR A Bit Description (cont)

Bit	Name	Status	Function
02	SYNCM A	Read/write	This bit is cleared on power-up or by writing a zero. When cleared, the UPD 7201 channel A transmitter and receiver both receive a clock derived from the programmable baud rate generator zero. When set, the transmitter receives a clock from the line (CCITT114) and the receiver receives a clock from the line (CCITT115). This bit effectively controls the internal baud rate or external baud rate selection.
01	SLU2B R EN	Read/write	This bit is cleared on power-up or by writing a zero. When cleared, the "party line receive data" to the UPD7201 channel B comes from the party line receiver if a jumper is selecting the party line receiver rather than the RS449 receiver. When written to a one, the party line receive data to channel B is disabled. This facilitates utilization of the PART LINE option.
00	SYNCM B	Read/write	This bit is cleared on power-up or by writing a zero. When cleared, the UPD7201 channel B transmitter and receiver receive a clock derived from the programmable baud rate generator zero. When set, the transmitter receives a clock from the line (CCITT114) and the receiver receives a clock from the line (CCITT115). This bit effectively controls the internal baud rate or external baud rate selection.

5.2.2 KXTCSR B Register

Figure 5-2 shows the bit positions and the bit definitions of the KXTCSR B register at address 177522. Table 5-2 describes the KXTCSR B register.

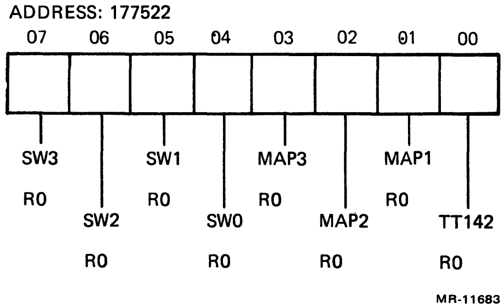


Figure 5-2 KXTCSR B Register

Table 5-2 KXTCSR B Bit Description

Bit	Name	Status	Function
07–04	SW3–SW0	Read only	These four hex-encoded general-purpose flags are controlled by a manual 16-position switch (boot/selftest switch) and can be utilized to cause different operational modes like autoboot, run selftest only, or enter ODT. Switch settings 7 and 15 are reserved for Digital use only and must not be selected. If console SLU1 Break is configured to cause a T-11 restart trap, the least significant 3 bits are forced to an all ones state until the RBUF is read to clear the condition. If this break trap in not configured, then the switch is not altered by a break condition. The native firmware will remain in ODT after a restart with a 7 or 15 seen until directed otherwise. Switch settings 8–15 will force MAP 3 bit to the zero state, which causes the RAM to be mapped in low address space.
03	MAP3	Read only	This bit reflects the state of a switch input to the local address decoder, which, when read as zero, selects RAM to be mapped starting at 000000, or, when read is asserted, selects RAM above PROM as determined by bits 02 and 01. Memory configurations are described in Chapter 6.
02–01	MAP2-MAP1	Read only	These bits reflect the state of two switches which configure the logical decode size of the four 28-pin user socket sites. Chapter 6 defines the four possible combinations.
00	TT142	Read only	When this bit is read in the zero state, the UPD 7201 channel A modem is not in a test mode and is available for normal service. When read in the one state, it indicates that the modem has been placed in a test condition. This condition is in response to an "on" condition of CCITT140 or CCITT141 and indicates that a test condition has been established.

5.2.3 KXTCSR C Register

Figure 5-3 shows the bit positions and the bit definitions of the KXTCSR C register at address 177524. Table 5-3 defines the KXTCSR C register.

NOTE

The high byte of registers KXTCSR A, KXTCSR B and KXTCSR C must be treated as indeterminate read and write data. These registers are low byte and word addressable by the T-11.

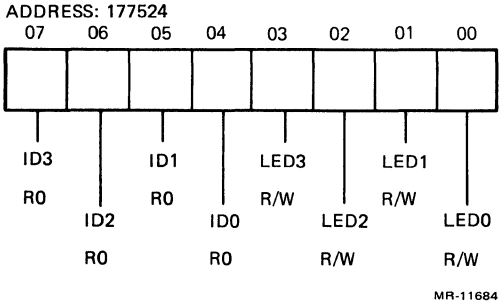


Figure 5-3 KXTCSR C Register

Table 5-3 KXTCSR C Bit Description

Bit	Name	Status	Function
07–04	ID3–ID0	Read only	These four bits reflect the SBC ID number assigned by the hex-encoded switch position. Codes 0 and 1 cause the Q-Bus interface to be disabled. Codes 2–7 indicate normal system operation with a base address in the lower quarter of the I/O page. Codes 8–15 indicate a base address in the upper quarter of the I/O page.
03–00	LED3–LED0	Read/write	These bits are cleared on power-up or by writing a zero. When in the zero state, these bits drive four red LEDs to the off state. Writing ones will turn the LEDs on. On power-up, the LEDs are on until the 8255 chip mode is programmed; then, the LEDs are turned off.

5.2.4 KXTCSR D Register

Figure 5-4 shows the bit positions and the bit definitions of the KXTCSR D register at address 177530 and Table 5-4 describes the register.

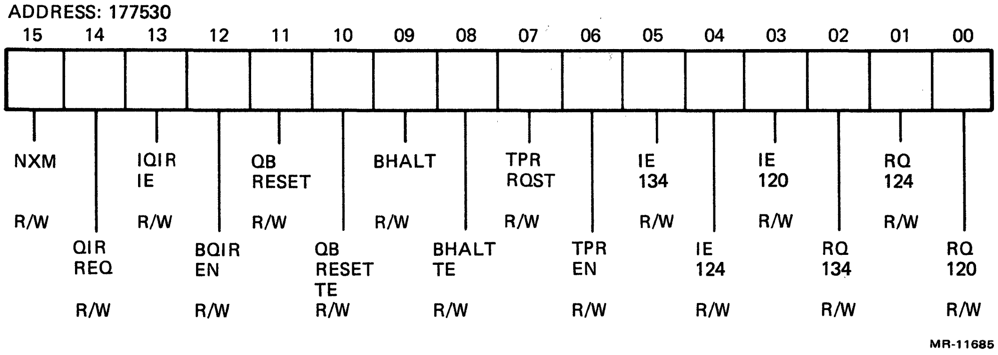


Figure 5-4 KXTCSR D Register

Table 5-4 KXTCSR D Bit Description

Bit	Name	Status	Function
15	NXM	Read/write	This bit is cleared on power-up, by a local RESET, or by writing a zero. This bit is set when the non-DMA local active address does not map to any local device and causes the T-11 to trap to the restart address if the T-11 is in control of the bus. If the DTC is in control of the bus, an EOP to the DTC is caused and this NXM bit is not set for Q-Bus or local accesses.
14	QIR Request	Read/write	This bit is cleared on power-up, by a local reset, or by a Q-Bus reset, by the reception of BIAKI, or by writing a zero. When set, this bit indicates that the Q-Bus interrupt register (QIR) has been written by the local processor, and that a Q-Bus interrupt request is pending service (i.e., BIRQ4 is asserted). If KXTCSR D bit 12 is set except when SBC ID numbers zero or one are selected, BIRQ4 is blocked. This bit cannot be set if the backplane is unterminated; it can be set if the backplane is terminated. Writing this bit to zero after it has been set by a write to the QIR flushes the pending request to the Q-Bus.

Table 5-4 KXTCSR D Bit Description (cont)

Bit	Name	Status	Function
13	IQIR IE	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero. When cleared the T-11 will not be interrupted when the QIR is emptied. When set by writing a one, the T-11 will receive a level 5 interrupt request for vector 130 when the QIR is serviced by the reception of BIAKI and the vector is taken, or when a Q-Bus reset occurs while the QIR is full, or when the QIR request bit is written to zero after having been set by a write to the QIR. Clearing this bit does not flush a pending Q-Bus interrupt request; it disables further T-11 interrupt requests. This bit does not disable the assertion of the Q-Bus BIRQ4 line.
12	BQIR EN	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero. When cleared, the QIR request is blocked from the Q-Bus. When set by writing a one, the Q-Bus will receive a level 4 interrupt request for the vector in the QIR. Clearing this bit does not flush a pending Q-Bus interrupt request; it blocks the request from the Q-Bus.
11	QB RESET	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero. It is set by the reception of Q-Bus BINIT if KXTCSR D bit 10 is set and causes a T-11 trap to 24. Once set, this bit blocks further traps to 24 until it is cleared. This bit should not be set if ID 0 or 1 is selected since BINIT may always be asserted in an unterminated backplane. For this case, no traps will occur if the bit is set.
10	QB RESET TE	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero. It is set by writing a one to enable the detection of a Q-Bus reset occurrence, which, when detected, sets KXTCSR D bit 11 and trap the T-11 to 24. Clearing this enable bit does not alter KXTCSR D bit 11; it prevents further trap requests.

Table 5-4 KXTCSR D Bit Description (cont)

Bit	Name	Status	Function
09	BHALT	Read/write	This bit is cleared .on power-up, by a local reset, or by writing a zero. It is set by the assertion of the Q-Bus BHALT signal when KXTCSR D bit 08 is set to flag the occurrence of this signal and cause a T-11 restart trap. Once set, further occurrences of BHALT are ignored, but other T-11 trap sources can still operate.
08	BHALT TE	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero. When clear, this bit prevents the SBC from responding to the assertion of BHALT. When written with a one, the assertion of the Q-Bus BHALT will cause a T-11 restart and set KXTCSR D bit 9. BHALT does not affect the DMA operation.
07	DPR RQST	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero. Writing a one has no effect. This bit reflects the state of the two-port register file (TPR) nonmaskable request bit which is set by any Q-Bus write to the first word of the two-port register file. Setting this bit causes a T-11 restart. Once set, further writes to the TPR first word (register zero) will not cause further T-11 restarts.
06	DPR EN	Read/write	This bit is cleared on power-up, by a local reset, or by writing a zero to force the Q-Bus processor to read zeros from the TPR. Q-Bus writes will timeout. When written with a one, the Q-Bus can read the actual contents of the TPR, and writes to the fifth, ninth, or thirteenth file words that will latch requests into KXTCSR D bits 00, 01, and 02. If KXTCSR D bits 03, 04, or 05 are set, level 5 interrupt requests are sent to the T-11. The T-11 or the DTC can always read the TPR, regardless of the state of this bit.

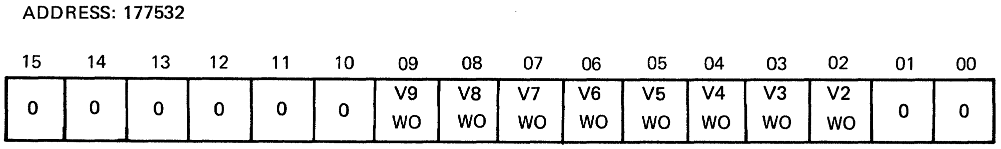
Table 5-4 KXTCSR D Bit Description (cont)

Bit	Name	Status	Function
05,04,03	IE134,124,120>	Read/write	These bits are TPR level 5 interrupt enables which are cleared on power-up, by local reset, or by writing zeros to block interrupt requests from the thirteenth, ninth, and fifth TPR on level 5 from the T-11. When these bits are set by writing ones, any pending or succedent level 5 vector 134, 124, or 120 interrupts can pass through to the T-11.
02,01,00	RQ134,124,120>	Read/write	These bits are TPR request flags which are cleared on power-up, by local reset, by writing zeros, or by interrupt acknowledge transactions. They are set by writes to the thirteenth, ninth, or fifth two-port registers from the Q-Bus to flag the T-11 of the occurrence of a write. If KXTCSR D bits 05, 04, or 03 are set, a T-11 level 5 interrupt will occur.

5.3 Q-BUS INTERRUPT REGISTER (QIR)

The local processor must signal the Q-Bus arbiter that it requires attention or that it has completed some task. The mechanism supporting this function is a word-addressable write-only vector register located at local address 177532 (see Figure 5-5). The local processor can write vector values between 0 and 1774 into this register, causing KXTCSR D bit 14 to set and the BIRQ4 signal to assert if the BQIR EN bit is set. The contents of this register are driven onto the Q-Bus when BIAKI is received and bits 14 and 12 in KXTCSR D are set. After the vector is transferred, the KXTCSR D status bit 14 is cleared. The QIR is not initialized on power-up nor is it cleared by a local reset or by an Q-Bus interrupt acknowledge. The register is updated with each write. If KXTCSR D bit 14 is set when a Q-Bus or a local reset occurs, the BIRQ4 driver will be disabled and the pending request killed along with KXTCSR D bit 14. If bit 13 of KXTCSR D was set, an interrupt is generated to the T-11 through vector 130 when the QIR is emptied by a Q-Bus IACK, a Q-Bus reset, or a T-11 write of the QIRRQ bit to zero. A local reset flushes the request, the interrupt enable, and the vector 130 interrupt.

When this register is driven onto the Q-Bus during an interrupt acknowledge transaction, the vector is continued in bits 02–09; all other bits are read by the processor as zeros.



ADDRESS: 177532

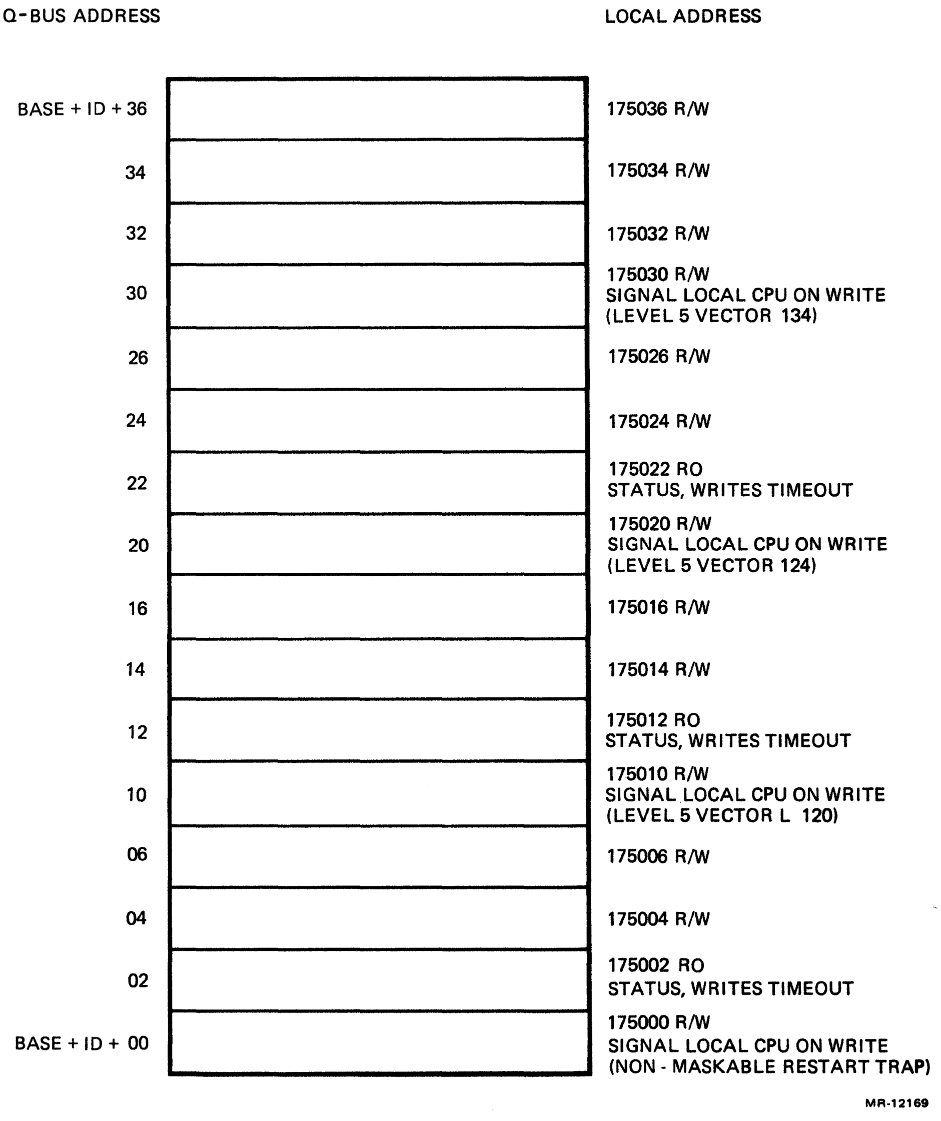
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	V9 WO	V8 WO	V7 WO	V6 WO	V5 WO	V4 WO	V3 WO	V2 WO	0	0

MR-11686

Figure 5-5 Q-Bus Interrupt Register (QIR)

5.4 TWO-PORT RAM (TPR)

The two-port RAM (TPR) contains 16 (word) registers with 16 bits in each. Both the local TDAL and global Q-Bus can do read/write operations via the TPR. The registers have a fixed local set of addresses (175016–175020, 175016–175000) and a Q-Bus address that is a variable base address plus 2. (See Figure 5-6). The first four registers, starting at register 0, are used exclusively to enable the Q-Bus processor to communicate with the SBC's native firmware. For more details on overall register usage and definitions, refer to Chapters 3 and 4.



Q-BUS ADDRESS	LOCAL ADDRESS
BASE + ID + 36	175036 R/W
34	175034 R/W
32	175032 R/W
30	175030 R/W SIGNAL LOCAL CPU ON WRITE (LEVEL 5 VECTOR 134)
26	175026 R/W
24	175024 R/W
22	175022 RO STATUS, WRITES TIMEOUT
20	175020 R/W SIGNAL LOCAL CPU ON WRITE (LEVEL 5 VECTOR 124)
16	175016 R/W
14	175014 R/W
12	175012 RO STATUS, WRITES TIMEOUT
10	175010 R/W SIGNAL LOCAL CPU ON WRITE (LEVEL 5 VECTOR L 120)
06	175006 R/W
04	175004 R/W
02	175002 RO STATUS, WRITES TIMEOUT
BASE + ID + 00	175000 R/W SIGNAL LOCAL CPU ON WRITE (NON - MASKABLE RESTART TRAP)

Figure 5-6 16 Bit × 16 Word Two-Port Ram (TPR)

5.4.1 Control Command Register (TPR 0)

The first register in the file is register 0 at local address 175000. There are six control commands that can be written into this register from the Q-Bus processor using the appropriate bit pattern. The control command register selects either a control command or test command. Bit 15 of the register selects which set of commands are used.

5.4.1.1 Control Commands – Figure 5-7 shows the bits used for all six control commands. All other bit combinations with bit 15 set to zero are considered erroneous. The SBC responds by setting the command error bit 15 in status register 1, clears the control command register, and enters the SBC into the wait state. Only one control command bit can be set at one time. Paragraph 4.4.1 begins the description of the function for each bit in this register.

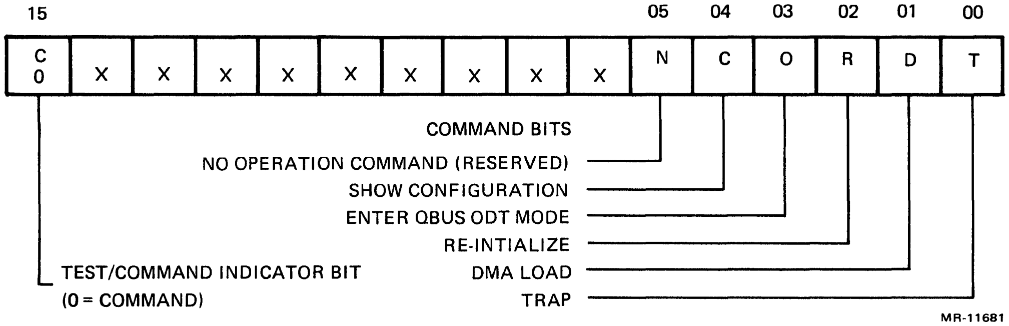


Figure 5-7 TPR Control Command Register 0

5.4.1.2 Test Commands – Figure 5-8 shows the bit pattern for all 11 test commands. Descriptions of all 11 test commands are included in Paragraph 4.4.2. As in the control command register, only one test selection bit can be set at one time. On completion of the test, the results are placed in TPR register 2 (local address 175004), register 3 (local address 175006), and the command register is cleared. (See Figure 5-6 and Figure 5-7). Execution of any of these tests may alter the state of the CSRs and RAM. All tests involve the use of interrupts only when the native RAM is mapped to location 0. This can be done by either setting the memory map configuration to a position which maps the native RAM low or by setting the boot/selftest switch to position 10. The SBC then enters the wait state on completion of the test. See Chapter 6 for memory maps.

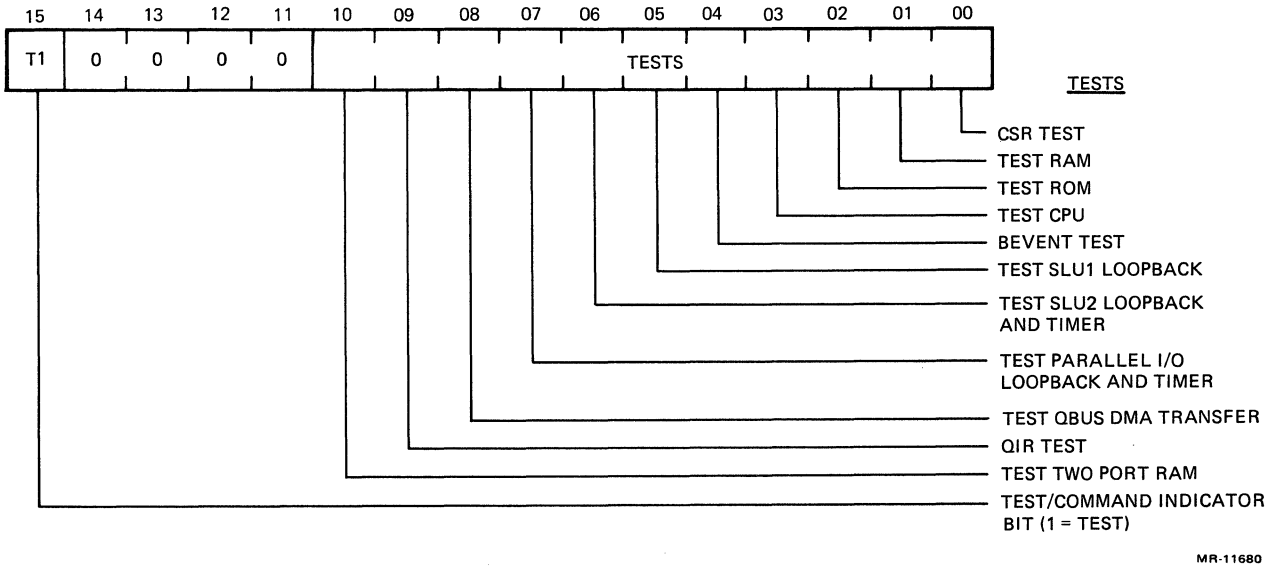


Figure 5-8 Test Command Register

5.4.1.3 ODT Format – After entering the Q-Bus ODT mode via the enter ODT command the format of the control register changes as shown in Figure 5-8. The rule for issuing Q-Bus commands are the same as for issuing control commands. This includes how the SBC responds to a command error. For definitions of each ODT command see Paragraph 4.4.3.

When the SBC is in the ODT mode, the internal readable microprocessor general-purpose registers are presented in register 2 by the following codes. All other codes are invalid.

Register	Code
R0	000000
R1	000001
R2	000002
R3	000003
R4	000004
R5	000005
SP	000006
PC	000007
PS	000010

5.4.2 Status Register (TPR 1)

The second TPR register is the SBC status register. Figure 5-9 shows the status register bit format. The following describes the bits as they pertain to the user.

NOTE
The bits marked for internal use are not to be altered by the user's application code.

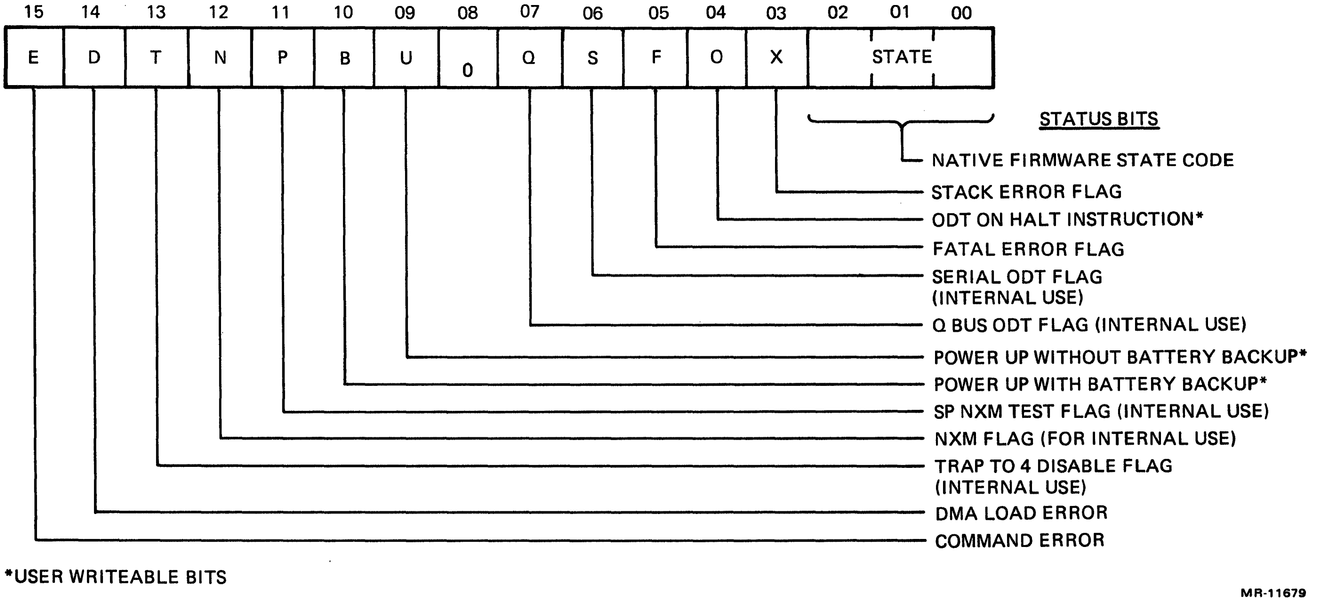

15
14
13
12
11
10
09
08
07
06
05
04
03
02
01
00
E
D
T
N
P
B
U
0
Q
S
F
O
X
STATE
STATUS BITS
NATIVE FIRMWARE STATE CODE
STACK ERROR FLAG
ODT ON HALT INSTRUCTION*
FATAL ERROR FLAG
SERIAL ODT FLAG (INTERNAL USE)
Q BUS ODT FLAG (INTERNAL USE)
POWER UP WITHOUT BATTERY BACKUP*
POWER UP WITH BATTERY BACKUP*
SP NXM TEST FLAG (INTERNAL USE)
NXM FLAG (FOR INTERNAL USE)
TRAP TO 4 DISABLE FLAG (INTERNAL USE)
DMA LOAD ERROR
COMMAND ERROR
*USER WRITEABLE BITS
MR-11679

Figure 5-9 TPR Status Register 1

1. Bits 00–02 are the state field of the status register. These bits are used for SBC status read by the Q-Bus arbiter. These bits cannot be written to by SBC application code or through ODT commands. The following describes the codes for each bit.

   a. State 000 – This field is read when the SBC is not available and occurs when the SBC is in serial ODT mode or during power-up or reinitialization. When in this state, no commands from the Q-Bus should be sent.

   b. State 001 – Indicates the SBC is performing selftests. This state should be brief unless a fatal error is encountered during the tests. If a fatal error is found, the SBC will remain in this state until power-down.

   c. State 010 – Indicates that the SBC boot/selftest switch is in position 10. This state is a dedicated test mode. The SBC should not receive commands to load or run application code.

   d. State 011 – Indicates that only Q-Bus commands are responded to by the SBC.

   e. State 100 – Indicates that the SBC is in an idle state and is waiting for a command from the Q-Bus arbiter. The SBC is in this state if the boot/selftest switch is set for not booting mode or after the following commands are received:

      - DMA load
      - Show configuration
      - Any test command
      - Any erroneous command

   f. State 101 – Indicates that the application code is being loaded or is attempting to load a boot block from a TU58 boot device connected to the serial console line (SLU1).

   g. State 110 – This is an invalid state indicating possible errors in this register.

   h. State 111 – Indicates that the SBC is passing control to RAM code or user ROM.

2. Bit 03 is the stack error flag. The flag is set when a trap to 4 emulation found the user stack pointing to a nonwritable memory location. This is necessary because the trap emulation involves the use of the stack. If the flag is set, the native firmware sets the stack pointer to the default user stack value (TOR – 100) before emulating the trap to four. The stack error bit is cleared when a trap to 4 emulation (without stack pointer problems) is performed. The application trap 4 handler may use this bit to determine if there was a stack problem and take appropriate action.

3. Bit 04 is set by the user if the SBC is to enter ODT in the execution of a halt instruction. If this bit is left cleared (cleared during initialization), the SBC responds to a halt instruction by performing a trap to 10 emulation.

4. Bit 05 is the fatal error flag. When this bit is set the SBC has detected a fatal error while executing a selftest or a Q-Bus controlled selftest. Also, this bit is set if the boot/selftest switch is set to an invalid setting. When the bit is set, the SBC is inoperative and does not respond to any Q-Bus commands. A BREAK command from the console serial interface (if the board is jumpered for BREAK enable) causes the SBC to enter serial ODT if it can.

5. Bits 06, 07, 11, 12 and 13 are for internal use by the native firmware. These bits should *never* be altered by the application code or through ODT.

6. Bit 09 is set on power-up if the SBC is not powering up from a battery backup state (see bit 10 description).

7. Bit 10 is set when the SBC powers up from a battery backup state. If bits 08 or 09 are to be used by the application code, they must be cleared after being tested. In this way, when a power-fail trap to 24 occurs, both bits are cleared, indicating that this is a power-fail interrupt. It is important to note that the SBC hardware traps to 24 on power-fail. In a MicroPower application the kernel handles these bits. These bits are user alterable. Table 5-5 summarizes the meaning of these bits.

8. Bit 14 is the DMA error bit that is described in the DMA application load description (Chapter 4) and bit 15 is the command error bit also described in Chapter 4.

Table 5-5 Power-Up Test Status

Bit* **10**	 **9**	**Description**
0	1	Power-up no battery backup
1	0	Power-up with battery backup
0†	0	Power-fail interrupt

* Bits 9 and 10 are both user writable bits.

† Bits are not altered by the native firmware. They will be 0 only if the application has previously cleared them.

5.4.3 Control Register 2 (TPR 2)

TPR register 2 is used by the native firmware to report errors encountered when the SBC performs any selftests. See Figure 5-10 for register 2 bit format. Only the bit in register 2 associated with the test run is set or cleared after the test is completed. This represents an accumulated error status format for handling more than one test error. Register 2 bits set to one indicate an error; bits set to zero indicate successful completion of the test.

5.4.4 Control Register 3 (TPR 3)

TPR control register 3 (local address 175006) is used to invoke, if necessary, a particular selftest. The selftest codes and discrete error codes passed are from the last test completed. See Figure 5-11 for register 3 bit format. The discrete error bits indicate which portion or portions of a selftest failed and test coded bits identify the test that was executed. Tables 5-6 through 5-15 describe each error bit for all 11 tests.

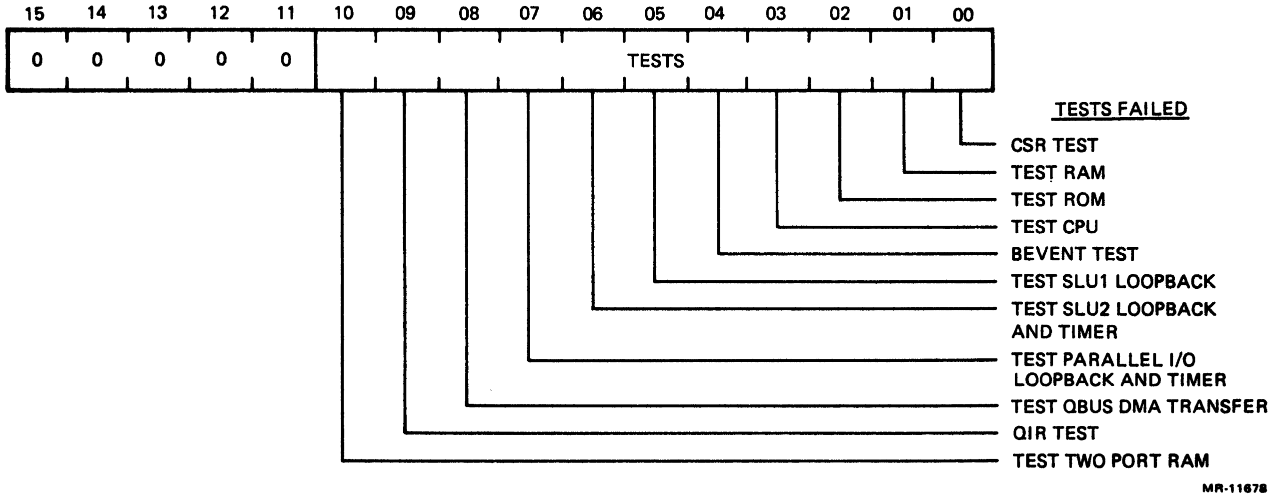


Figure 5-10 TPR Control Register 2

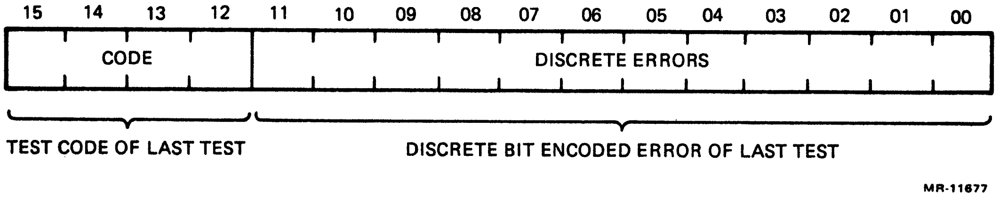


Figure 5-11 TPR Control Register 3

Table 5-6 CSR Test Status (Test Code = 01)

Bit	Location	Local Address
00	CSR (control status register)	177530
01	QIR (Q-Bus interrupt register)	177532
02	TPR	175000–175016 175020–175036
03	SLU1 Driver	177566–177560
04	SLU2 Driver	175736–175700
05	Counter/Timer	175736
06	PI/O Counter/Timer	177136–177000
07	DTC	174536–174400

Table 5-7 RAM Test Status (Test Code = 02)

Bit	Error
00	Native RAM address is nonexistent memory (NXM)
01	Native RAM write-read
02	Native RAM read-modify-write (RMW)
03	Native RAM read-modify-write to low byte
04	Native RAM read-modify-write to high byte
05	User RAM address is nonexistent memory
06	User RAM write-read
07	User RAM read-modify-write (RMW)
08	User RAM read-modify-write to low byte
09	User RAM read-modify-write to high byte

Table 5-8 CPU Test Status (Test Code = 04)

Bit	Error
00	CPU test failed

Table 5-9 BEVNT Test Status (Test Code = 05)

Bit	Error
00	No RAM in vector space
01	Clock interrupt not masked by level 6
02	Clock does not interrupt
03	Clock cannot be disabled

Table 5-10 SLU1 Test Status (Test Code = 06)

Bit	Error
00	No RAM in vector space
01	Transmit interrupt not masked at level 4
02	Transmit interrupt not received
03	Receive interrupt not masked at level 4
04	Receive interrupt not received
05	Receive data not correct
06	Receive done failed

Table 5-11 SLU2 Test Status (Test Code = 07)

Bit	Error
00	No RAM in vector space
01	Interval timer does not interrupt
02	Async mode data transfer incomplete
03	Sync mode EOF-SDLC not received
04	Sync mode data transfer not complete
05	Sync/async modes receive data incomplete
06	DMA mode data transfer incomplete
07	DMA mode received data incomplete
08	Status incorrect or no interrupt request with "request to send" set
09	Status incorrect or no interrupt request with "RS-422" set
10	Status incorrect or no interrupt request with "terminal in service" set
11	Status incorrect with everything off

Table 5-12 Parallel I/O Test Status (Test Code = 10)

Bit	Error
00	No RAM in vector space
01	Reset state incorrect
02	Timer does not start
03	Timer does not stop
04	Interrupt not masked by level 4
05	Timer interrupt not received
06	Loopback timeout
07	Receive data incorrect

Table 5-13 DMA Test Status Test Code = 11

Bit	Error
00	No RAM in vector space
01	Q-Bus address undefined (for local test only)
02	Chain interrupt not received
03	DMA channel hung up
04	DMA aborted
05	DMA data error

Table 5-14 Q-Bus Interrupt Register Test Status (Test Code = 12)

Bit	Error
00	Q-Bus vector undefined
01	Break request was not set
02	No RAM in vector space
03	BIACK interrupt not masked at priority level 5
04	BIACK interrupt not received
05	BIACK not cleared by break request
06	Bus request not received

Table 5-15 Two-Port RAM (TPR) Test Status (Test Code = 13)

Bit	Error
00	Local TPR write-read
01	Bus error on Q-Bus TPR read
02	Non-zero data on Q-Bus read
03	No RAM in vector space
04	Bus error on Q-Bus TPR write
05	No interrupt on write to command register
06	Q-Bus write or interrupt with TPR disabled
07	Q-Bus write did not time out
08	Q-Bus write timed out
09	TPR interrupt level 4 not received
10	TPR interrupt level 8 not received
11	TPR interrupt level 12 not received

5.4.5 ROM Test Status (Test Code = 03)

The test performs a checksum of the native ROM contents and the user ROM if selected. If user ROM is installed, TPR register 3 is set for 0000001 when the ROM test is selected. When the user ROM test is not selected TPR register 3 must be set for 0000000. The algorithm for the ROM test follows.

```
SET CHECKSUM = 0
DO UNTIL ALL ROM CHIP LOCATIONS TESTED
        ADD NEXT BYTE TO CHECKSUM (BYTE)
        SHIFT CHECKSUM LEFT 1 (BYTE)
        ADD CARRY INTO CHECKSUM
END UNTIL
```

This checksum test is passed if the difference of the above calculation and the checksum number of the ROM being tested is equal to zero. The resultant checksum number is placed in the last physical location of each ROM.

Prior to entering the DMA controller test, register 3 must be loaded with the high 16 bits from the base address of the arbiter RAM 4 K words to be used. If, during this interval, a zero is written into TPR 3, only the local portion of the test is performed. The Q-Bus portion is disabled.

5.5 DMA TRANSACTIONS

The controller registers are categorized into chip-level registers, which control the overall operation and configuration of the DMA controller, and channel-level registers, which are duplicated for each channel. The two chip-level registers are the master mode register and the command register. All registers must be treated as word-wide during programmed transactions on even address boundaries. DMA controller registers are addressable at local I/O page addresses 174536–174400. Table 5-16 lists the registers by their address, provides the number of bits used in the registers, and provides the register name.

The following paragraphs (5.5.1 through 5.5.6) detail the bit assignments listed in Table 5-16 and provide a set of programming rules for the manipulation of these bits, which if followed will ensure correct controller chip behavior. Certain operational modes of the DMA controller (DTC) are not supported and the following paragraphs describe these cases as "must be" or "illegal" conditions.

Table 5-16 DTC Internal Registers

Address	Access	Used Bits	Channel	Register Name
174400	R/W/C*	16	CH1	Current B address offset field
174402	R/W/C	16	CH0	Current B address offset field
174404	R/W/C	16	CH1	Base B address offset field
174406	R/W/C	16	CH0	Base B address offset field
174410	R/W/C	16	CH1	Current A address offset field
174412	R/W/C	16	CH0	Current A address offset field
174414	R/W/C	16	CH1	Base A address offset field
174416	R/W/C	16	CH0	Base A address offset field
174420	R/W/C	16	CH1	Current B address segment/tag field
174422	R/W/C	16	CH0	Current B address segment/tag field

Table 5-16 DTC Internal Registers (cont)

Address	Access	Used Bits	Channel	Register Name
174424	R/W/C	10	CH1	Base B address segment/tag field
174426	R/W/C	10	CH0	Base B address segment/tag field
174430	R/W/C	16	CH1	Current A address segment/tag field
174432	R/W/C	16	CH0	Current A address segment/tag field
174434	R/W/C	16	CH1	Base A address segment/tag field
174436	R/W/C	16	CH0	Base A address segment/tag field
174440	R/W/C	16	CH1	Chain address offset field
174442	R/W/C	16	CH0	Chain address offset field
174444	R/W/C	10	CH1	Chain address segment/tag field
174446	R/W/C	10	CH0	Chain address segment/tag field
174450	R	16	CH0	Interrupt save register
174452	R	16	CH0	Interrupt save register
174454	R/W	16/8	CH1 and CH0	CH1 status/CH0 and CH1 command
174456	R	16	CH0	CH0 status
174460	R/W/C	16	CH1	Current operation count
174462	R/W/C	16	CH0	Current operation count
174464	R/W/C	16	CH1	Base operation count
174466	R/W/C	16	CH0	Base operation count
174470	R/W	8	CH0 and CH1	Master mode
174472	–		X†	
174474	–		X	
174476	–		X	
174500	–		X	
174502	–		X	
174504	–		X	
174506	–		X	
174510	R/W/C	16	CH1	Pattern
174512	R/W/C	16	CH0	Pattern
174514	R/W/C	16	CH1	Mask
174516	R/W/C	16	CH0	Mask
174520	R/W/C	16	CH1	Channel mode low
174522	R/W/C	16	CH0	Channel mode low
174524	R/W/C	5	CH1	Channel mode high
174526	R/W/C	5	CH0	Channel mode high
174530	R/W/C	8	CH1	Interrupt vector
174532	R/W/C	8	CH0	Interrupt vector
174534	–		X	
174536	–		X	

* R = local processor can read this register
W = local processor can write this register
C = chain operations can load this register
† X = unused registers; reads produce indeterminate data, writes have no effect

5.5.1 Master Mode Register (MM)

This 8-bit read/write register (local address 174470) is cleared to all zeros on power-up, by a T-11 reset instruction, or by issuing a DTC reset command. (See Figure 5-12.) The bits and their functions follow.

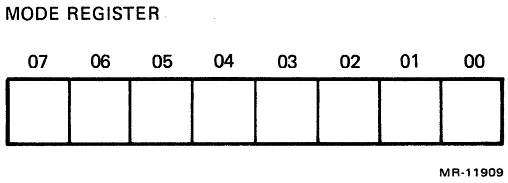


Figure 5-12 DTC Master Mode Register

Bit 07 – This bit must always be written to zero.

Bit 06 – This bit must always be written to one to enable a vector to be sent during an interrupt acknowledge transaction.

Bit 05 – This bit must always be written to zero to enable the interrupt save register to be driven onto the bus during interrupt acknowledge transactions.

Bit 04 – This bit can be written to a one, preventing an interrupt request from being asserted by the counter timer chip which is on an interrupt daisy chain with the DTC at a lower hardware level. When cleared, the counter timer chip can interrupt the T-11. This bit is known as the disable lower chain control bit.

Bit 03 – This bit must always be set to one, enabling asynchronous controller operation when the Q-Bus or TDAL bus is a source or a destination. It may be set to zero and programmed wait states can be used for certain slow local devices (writes to the 8255A-5, ZCIO, or UPD 7201).

Bit 02 – When set to a one, the controller interleaves bus control with the local processor. When clear, the controller retains bus control until a terminal condition is reached by the operation count register if hog mode is selected.

Bit 01 – This bit must be written to zero.

Bit 00 – When written to a one, the controller can request bus control and perform operations. When cleared to zero, the controller is inhibited from requesting the bus and, consequently, from performing operations.

5.5.2 COMMAND Register (CM)

This 8-bit write-only register (local address 174454) is written to by the local processor to start execution of controller commands immediately after they are written. The register is divided into four fields: a 3-bit function field, a 3-bit interrupt control field, a 1-bit set/clear function control field, and a 1-bit channel select field. Table 5-17 lists the commands and the octal value which represents the command.

The reset command causes the controller to return to an initial state with MM register zeroed (other registers are not altered) and channel activity inhibited. A T-11 reset instruction execution has the same

Table 5-17 Command Register Operation Codes

Command	Channel	Op Codes
Reset	–	000*
Start chain	CH0	240*
Start chain	CH1	241*
Set software request	CH0	102*
Set software request	CH1	103*
Clear software request	CH0	100*
Clear software request	CH1	101*
Set hardware mask	CH0	202*
Set hardware mask	CH1	203*
Clear hardware mask	CH0	200*
Clear hardware mask	CH1	201*
Set CIE, IUS, IP	CH0	076†
Set CIE, IUS, IP	CH1	077†
Clear CIE, IUS, IP	CH0	074†
Clear CIE, IUS, IP	CH1	075†
Set CIE	CH0/CH1	062/063†
Clear CIE	CH0/CH1	060/061†
CIE is not altered	CH0/CH1	Bit 4 = 0†
Set IUS	CH0/CH1	052/053†
Clear IUS	CH0/CH1	050/051†
IUS is not altered	CH0/CH1	Bit 3 = 0†
Set IP	CH0/CH1	042/043†
Clear IP	CH0/CH1	040/041†
IP is not altered	CH0/CH1	Bit 2 = 0†
Set flip bit	CH0	142*
Set flip bit	CH1	143*
Clear flip bit	CH0	140*
Clear flip bit	CH1	141*

* Commands which do not specifically utilize the interrupt control field cannot alter the field when they are issued but treat bits 4, 3, and 2 as "don't cares."

† Bits 4, 3, and 2 are enable operation control bits which, when set, will enable IE, IUS, and IP bits to be altered. When cleared, the IE, IUS, and IP bits are not altered.

effect as the DTC reset command. On power-up, however, the DTC is initialized, the register contents are undefined and must be written before a start command is issued. Or, the chain address can be written, a start chain command can be issued, and the selected channel clears the no auto-reload or chain bit in the channel's status register, which starts a chain reload operation of the channel's registers. The start chain channel command is only honored if both the chain abort (CA) and second interrupt pending (SIP) bits in the channel status register are clear. If either bit is set, the command is disregarded. When the waiting for bus bit in the status register is set, and a start chain command is issued, the channel honors the command after one DMA iteration. The software request command sets the software request bit in the selected channel's mode register. If the second interrupt pending (SIP) bit and no auto-reload or chain (NAC) bit in the channel's status register are both clear, the channel starts executing the programmed DMA operation. If either the SIP or NAC bit is set, the channel does not start executing a DMA operation until both bits are cleared. The SIP bit clears when the channel receives an interrupt acknowledge. One way to clear the NAC bit is to issue a start chain command to the channel.

If the fetched reload word is all zeros, the channel's registers remains unchanged and the software request bit, if set earlier by command, causes the programmed DMA operation to start immediately. If during chaining new information is loaded into the channel mode register, this new information overwrites the software request bit.

The set/clear hardware mask command sets or clears the hardware mask bit in the selected channel's mode register. This command always takes effect. The hardware mask bit inhibits recognition of an active signal on the channel's DMA request pin; this bit does not affect recognition of a software request.

The flip bit in the selected channel's mode register can be cleared and set by the flip bit command. This allows the source and destination to be reversed, thereby reversing the data transfer direction without reprogramming the channel. Chaining new information into the channel mode register overwrites the flip bit.

The set/clear channel interrupt enable (CIE), interrupt pending (IP), and interrupt under service (IUS) commands allows the user to either set or clear any combination of these bits in the selected channel's status register. These bits control the operation of the channel's interrupt structure. Setting the IP bit causes the interrupt save register to be loaded with the current vector and status. The IP and IUS bits can be simultaneously cleared to facilitate an efficient conclusion to the processing of an interrupt.

5.5.3 Chain Control Register (CC)

This register cannot be written to or read by the local microprocessor. When a channel starts a chaining operation, it fetches a reload word from the memory location pointed to by the chain address register and stores it in the CC register.

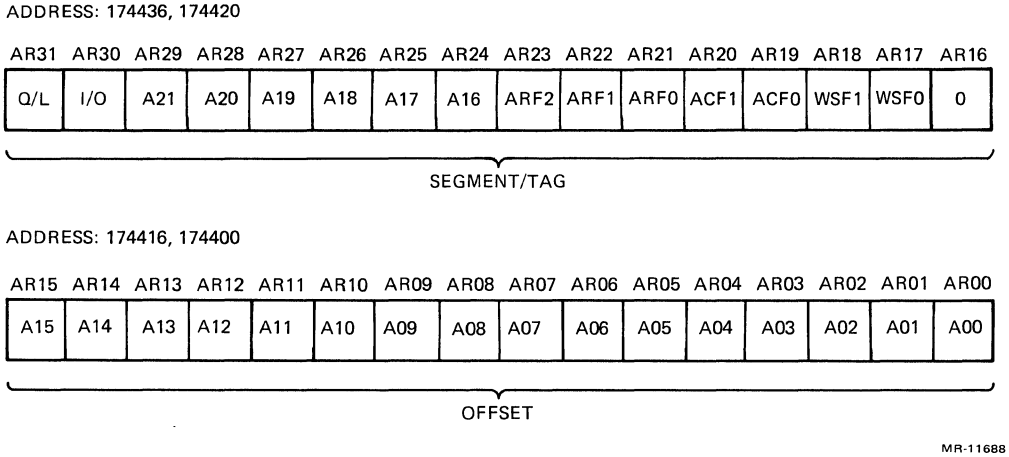
5.5.4 Temporary Register (TMP)

The temporary register is used to store data during flowthrough transfers and to hold data being compared during a search or transfer-and-search. The temporary register cannot be written to or read from by the local processor.

5.5.5 Current Address Registers

The current address registers A and B (current ARA and ARB) are used to point to the source and destination addresses for DMA operations. The base address registers A and B (base ARA and ARB) contents are transferred into the current ARA and ABA registers at the end of a DMA operation if the user enables base-to-current reloading in the completion field of the channel mode register. This facilitates DMA operations without reloading the current registers. The ARA and ARB registers can be loaded during chaining and can be written to or read by the local processor. Each of the base and current ARA and ARB registers consists of two words organized as a 7-bit tag field and an 8-bit segment in one word and a 16-bit offset in the other. The segment and offset contain the actual address driven onto the bus. The tag field selects whether the address is to be incremented, decremented, or left unchanged. It also selects status codes associated with the address. The tag field also allows the user to insert 0, 1, 2, or 4 wait states into memory or I/O accesses addressed by the offset and segment fields. These programmed wait states are added to the asynchronous wait states generated external to the DTC.

The address reference field portion of the tag field can contain ARF (000) or ARF (011) for all local or Q-Bus transactions. The 000 value enables the addresses to increment/decrement by one for byte operations and by two for word operations. The 011 value enables the address to increment/decrement by two for byte and word operations. Low byte, high byte, and word-to-byte funneling operations are supported locally only. Only word operations are supported across the Q-Bus interface. Word operations require that even address values be placed in the address registers. Low byte operations require that odd address values be placed in the byte address registers. High byte operations require that even values be placed in the byte address registers. Byte operations are only supported locally; attempted byte writes to the Q-Bus result in word operations with the unspecified byte being corrupted. Q-Bus word-to-local byte funneling is fully supported, as is local byte-to-Q-Bus word funneling. The 8-bit segment field contains the most significant address bits (16–21) and an address space select bit. The I/O page is decoded from the TDAL lines during local DMA operations to select a 16-bit I/O page locally. The Q-Bus I/O page bit selects either a 16-, 18-, or 22-bit I/O page on the Q-Bus. Bit AR31 determines whether a local or Q-Bus operation is selected. Figure 5-13 shows the ARA and ARB registers.



ADDRESS: 174436, 174420

AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16
Q/L	I/O	A21	A20	A19	A18	A17	A16	ARF2	ARF1	ARF0	ACF1	ACF0	WSF1	WSF0	0

SEGMENT/TAG

ADDRESS: 174416, 174400

AR15	AR14	AR13	AR12	AR11	AR10	AR09	AR08	AR07	AR06	AR05	AR04	AR03	AR02	AR01	AR00
A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

OFFSET

MR-11688

Figure 5-13 Address Registers A and B

The Q/L bit, when set to one, causes the address to reference the Q-Bus space; when cleared, the local space is accessed. The I/O bit controls the assertion of the Q-Bus BBS7 line during DMA transactions to Q-Bus space. The one state asserts BBS7. This bit has no affect during local references since physical addresses are decoded. Note that A21–A16 are ignored by the local I/O page decode logic when Q/L is detected cleared. The A21–00 bits correspond with address bits BDAL21–00 for Q-Bus space and TDAL15–00 for local space.

NOTE
For local references, A21--16 bits should be zeros.

For local low byte operations, the A00 bit must be one; for word operations, it must be zero. The ACF bits select one of three counting methods as follows.

00 = increment address
01 = decrement address
10 = hold address
11 = hold address

The WSF bits select the number of wait states to install when accessing local devices which are slow. These bits are encoded as follows.

00 = 0 wait states
01 = 1 wait states
10 = 2 wait states
11 = 4 wait states

Bits AR23, AR22, AR21, and AR16 of the tag field are not used and must be set to zero.

The 16-bit, current-operation count register is used to specify the number of words or bytes to be transferred, searched, or transferred-and-searched. For word-to-word operations, the current operation count register must be programmed with the number of words to be transferred or searched. Each time a data is transferred or searched, the operation count register is decremented by one.

Once all of the data is transferred or searched, the transfer or search operation stops, the current operation count register contains all zeros, and the TC bit in the status register is one.

If the transfer or search stops before the current operation count register reaches zero, the register contents will indicate the number of words remaining to be transferred or searched. This allows a channel – which has been stopped prematurely by the local processor, or a negation of the channel request line, or a nonexistent memory reference (NXM), – to be restarted where it left off without having to reload the current operation count register. It also allows the local processor to determine how many operations remain.

The maximum number of words to be transferred or searched are specified by setting a word count of zero. Care must be taken to ensure that operation counts with incrementing/decrementing addresses do not wrap over into the I/O page or under into the I/O page.

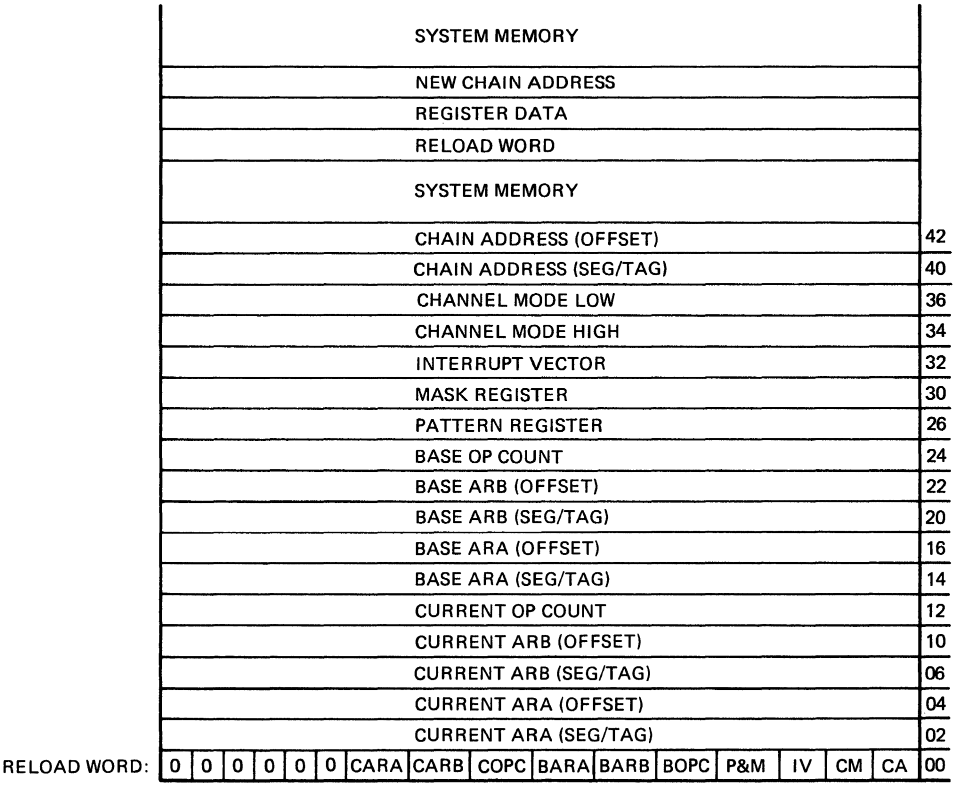
When a channel finishes a DMA operation, the user may choose to perform a base-to-current reload. In this type of reload, the current address registers A and B are loaded with the data in the base address registers A and B, respectively, and the current operation count register is loaded with the data in the base operation count register. The base-to-current reload operation facilitates repetitive DMA operations without the multiple memory accesses required by chaining. Although the channel must have local bus control to perform base-to-current reloading, complete reloading operation occurs in four clock cycles.

Note that if the channel had to relinquish the local bus control because two unacknowledged interrupts were queued, it would have to regain bus control to perform any base-to-current reloading (or chaining, for that matter). In this case, it acquires the bus once an interrupt acknowledge is received, even if it relinquishes the bus immediately afterward because no hardware or software request is present.

If the channel is programmed to chain at the end of a DMA operation, it will use the chain address register to point to a chain control table in memory. The first word in the table is a reload word, specifying the register(s) to be loaded. Following the reload word are the data values to be transferred into the register(s). Because chaining occurs after base-to-current reloading, it is possible to reset the current address registers A and B and the current operation count register to the values used for previous DMA operations and then chain reload one or two of these registers to some special value to be used, perhaps, for this DMA operation only. For burst operations of, say, four or eight words, the DTC is set up to interrupt the T-11 on end of process (EOP), to chain reload the operation count and channel mode register with the software request bit cleared, and to operate in demand dedicated with bus release mode.

The DTC will notify the T-11 after each operation terminates. The T-11 can then decide to either issue a software start command or terminate the process and set up a new process. If the base values are not reloaded during chaining, the channel can revert to the base values at a later cycle. If an all-zero-reload word is fetched during chaining, the chain operations do not reload any registers; but in all other respects, it performs like any other chaining operation. Thus, the chain address is incremented by two to point to the next word in memory, and at the end of the all-zero-reload word chain operation, the channel will be ready to perform a DMA operation. All-zero-reload words are useful to start or terminate linked lists of DMA operations traversed by chaining. On the other hand, care must be taken in their use since the channel may perform an erroneous operation if it is unintentionally started after the chaining operation.

The two-port register file supports 16 of the possible 18 reload parameters at one time, so it is possible for the Q-Bus arbiter to load the two-port register (TPR) with an operation and to have the controller pick information up and perform the operation. Then the controller returns to the two-port register for the next operation or waits until told to reload before returning to the two-port register for new parameter information. The controller can also go to Q-Bus memory or to local memory to pick up the parameter information as specified by the Q/L bit in the chain address register. The reload word format is shown in Figure 5-14.



SYSTEM MEMORY	
NEW CHAIN ADDRESS	
REGISTER DATA	
RELOAD WORD	
SYSTEM MEMORY	
CHAIN ADDRESS (OFFSET)	42
CHAIN ADDRESS (SEG/TAG)	40
CHANNEL MODE LOW	36
CHANNEL MODE HIGH	34
INTERRUPT VECTOR	32
MASK REGISTER	30
PATTERN REGISTER	26
BASE OP COUNT	24
BASE ARB (OFFSET)	22
BASE ARB (SEG/TAG)	20
BASE ARA (OFFSET)	16
BASE ARA (SEG/TAG)	14
CURRENT OP COUNT	12
CURRENT ARB (OFFSET)	10
CURRENT ARB (SEG/TAG)	06
CURRENT ARA (OFFSET)	04
CURRENT ARA (SEG/TAG)	02

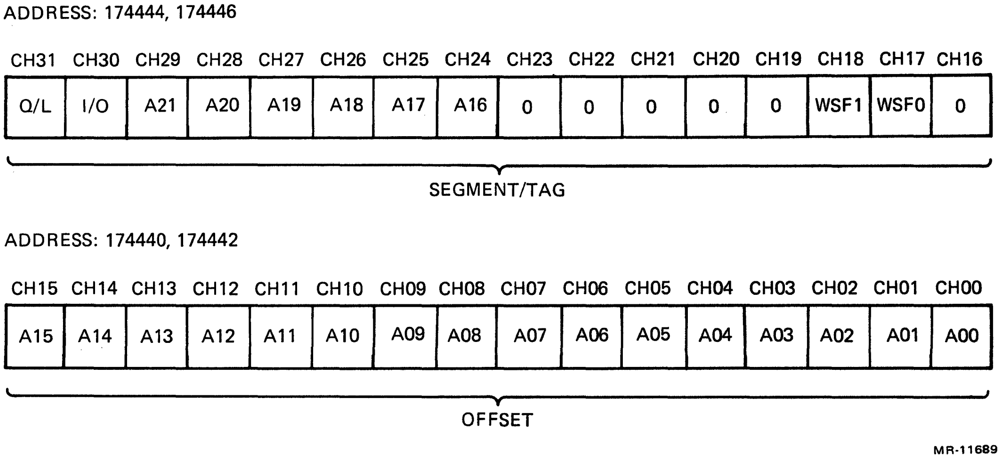
RELOAD WORD:	0	0	0	0	0	0	CARA	CARB	COPC	BARA	BARB	BOPC	P&M	IV	CM	CA	00

CA = CHAIN ADDRESS	BARB = BASE ARB
CM = CHANNEL MODE	BARA = BASE ARA
IV = INTERRUPT VECTOR	COPC = CURRENT OP-COUNT
P&M = PATTERN AND MASK	CARB = CURRENT ARB
BOPC = BASE OP-COUNT	CARA = CURRENT ARA

MR-11676

Figure 5-14 DTC Internal Registers

Each channel has a chain address register which points to the chain control table in memory containing data to be loaded into the channel's registers. The chain address register (Figure 5-15) is two words long. The first word consists of a segment and tag field. The second word contains the 16-bit offset portion of the memory address. The tag field contains two bits to designate the number of wait states to be inserted during accesses to the chain control table. The chain control table should reside in local space, but can reside in Q-Bus space.



ADDRESS: 174444, 174446

CH31	CH30	CH29	CH28	CH27	CH26	CH25	CH24	CH23	CH22	CH21	CH20	CH19	CH18	CH17	CH16
Q/L	I/O	A21	A20	A19	A18	A17	A16	0	0	0	0	0	WSF1	WSF0	0

ADDRESS: 174440, 174442

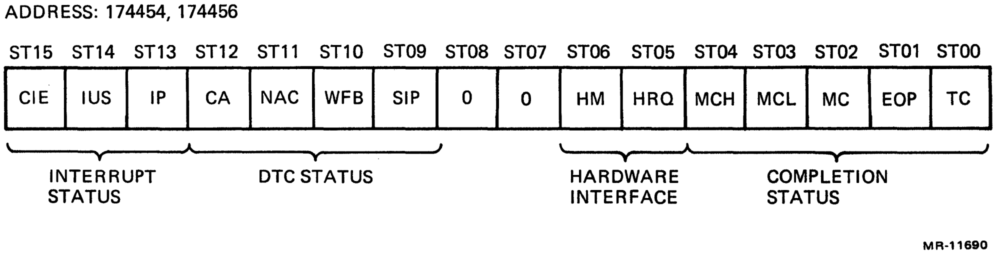
CH15	CH14	CH13	CH12	CH11	CH10	CH09	CH08	CH07	CH06	CH05	CH04	CH03	CH02	CH01	CH00
A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A00

Figure 5-15 Chain Address Register

When the Q-Bus space is selected, the wait states are generated external to the chip and programmed wait states are not required. The chain address register may be loaded during chaining and may be read from and written to by the local processor. If an NXM occurs during chaining, the chain address register holds the old address. This is true even if the access failure occurred while new chain address data was being loaded, since the old data is restored unless both words of the new data are successfully read. Note, however, that NXM's that occur when chaining and while loading a new chain address cause the new data to be lost.

The Q/L, I/O, WSF1, WSF0, and A21–00 lines are functionally the same as the corresponding lines in address register ARA and ARB described previously, except for the ACF and ARF fields, which are defaulted to zero by the DTC. The chain address register is always incremented by two and assumes memory data operations. The 0 bits are not assigned and are read as zeros and must be written with zeros. The A00 bit must be written to zero by all write or chain operations to this register.

The two 16-bit status registers (Figure 5-16) are read-only registers which report on the status of their associated channels. Each register has five fields: a 3-bit interrupt status field, a 4-bit DMA controller status field, a 2-bit reserved field, a 2-bit hardware interface status field, and a 5-bit completion status field. There is one status register for each channel.



ADDRESS: 174454, 174456

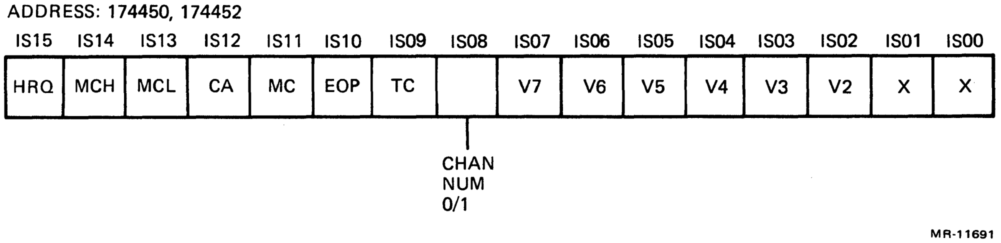
ST15	ST14	ST13	ST12	ST11	ST10	ST09	ST08	ST07	ST06	ST05	ST04	ST03	ST02	ST01	ST00
CIE	IUS	IP	CA	NAC	WFB	SIP	0	0	HM	HRQ	MCH	MCL	MC	EOP	TC

Figure 5-16 Status Register

Each controller channel is a source of interrupts and has three bits that control how the channel generates interrupts. The interrupt status bits are programmed from the command register. These bits – channel interrupt enable (CIE), interrupt pending (IP), and interrupt under service (IUS) – for each channel must be monitored in the interrupt service routine and initialized via the command register. They are also copied into the interrupt save register when the channel interrupt request is posted. To allow the controller to start executing a new operation after issuing an interrupt (but before an interrupt acknowledge is received), a two-deep interrupt queue is implemented on each channel. Each channel has its own interrupt vector register for identifying the source of the interrupt during the acknowledge transaction. While an interrupt source has an interrupt under service (IUS = 1), it prevents all lower priority interrupt sources from requesting interrupts. When interrupt servicing is complete, the local processor must reset the IUS bit by means of a command to the command register.

Once a channel issues a T-11 interrupt on maskable level 4, it is desirable to allow the channel to proceed with the next DMA operation before the interrupt is acknowledged. This could lead to problems if the controller channel attempted to chain reload the vector register contents. In such a situation, it may not be clear whether the old or new vector would be returned during the acknowledge. This dilemma is resolved in the controller by providing each channel with an interrupt save register. When the channel sets the IP as part of the procedure followed to issue an interrupt, the contents of the vector register and some of the status register bits are saved in an interrupt save register.

When an interrupt acknowledge cycle is performed, the contents of the interrupt save register (Figure 5-17) are driven onto the bus and only IS7-2 are read by the T-11 from TDAL 7–2. Although the use of an interrupt save register allows the channel to proceed with a new task, problems may occur if a second interrupt is to be issued by the channel before the first interrupt is acknowledged. To avoid conflicts between the first and second interrupt, each channel has a second interrupt pending (SIP) bit in its status register. When a second interrupt is to be issued before the first interrupt is acknowledged, the SIP bit is set and the channel relinquishes the bus until an acknowledge occurs. For compatibility with polled interrupt schemes, the interrupt save register can be read by the local processor without wait states. Whenever the IP bit is set, the interrupt save register is loaded from the vector and status register. Note that the SIP bit is transferred to the IP bit when IP is cleared by the local processor. Whenever IEI is high, CIE is set and IUS is cleared, the interrupt request line will assert as IP is set. Note that some care must be taken in programming the interrupt service routines to reset IUS if the user does not wish to reset IUS until both interrupts are serviced.


ADDRESS: 174450, 174452

IS15	IS14	IS13	IS12	IS11	IS10	IS09	IS08	IS07	IS06	IS05	IS04	IS03	IS02	IS01	IS00
HRQ	MCH	MCL	CA	MC	EOP	TC		V7	V6	V5	V4	V3	V2	X	X

Figure 5-17 Interrupt Save Register

The DTC status field reports on the current channel state. The "channel initialized and waiting for request" status is not explicitly stated; it is reflected by status bits ST12–09 being all zero. The waiting for bus (WFB) status bit (ST10), when set, indicates that the channel wants bus control to perform a DMA operation.

The channel may or may not be actually asserting the request line to the local processor, depending on the programming of the master mode chip enable bit and the state of the request line and the grant line when the channel decided it wanted the bus. If a channel completes a DMA operation and neither base-to-current reloading nor auto-chaining were enabled, the no auto-reload or chaining (NAC) bit will be set. The NAC bit is only reset when the channel receives a start chain command. If two interrupts are queued, the second interrupt pending bit (SIP) will be set and the channel will be inhibited from further activity until an interrupt acknowledge occurs. Finally, if the channel address is nonexistent during chaining, the chain abort (CA) and the NAC bit will be set. These bits are also set when the DTC is reset by command or hardware. The CA bit holds the NAC bit in the set state until the end of process (EOP) bit is cleared. The CA bit is cleared when a new chain address segment and tag word or offset word is loaded into the channel address registers.

The hardware interface field provides a hardware request (HRQ) bit which provides a means of monitoring the channel's hardware request line. When the request is asserted, the HRQ bit will be one. The HM bit, when set, prevents the controller from responding to the request line. Note, however, that the HRQ bit always reports the true (unmasked) status of the request pin. The HM bit can be cleared by software command.

When a DMA operation ends, the channel can interrupt the local processor, perform base-to-current reloading, chain reload the next DMA operation, perform any combination of the above, or take no action. The action to be performed by the channel is selected in the completion option field of the channel mode register. The completion field stores data at the end of each DMA operation. This data indicates why the DMA operation ended. When the next DMA operation ends, new data is loaded into these bits, overwriting and, thereby, erasing the old data.

Three bits indicate whether the DMA operation ended as a result of a terminal count (TC), a match count (MC), or an end of process (EOP) termination. The TC bit sets to one if the operation count reaching zero ended the DMA operation. The MC bit will be set to one if an MC termination occurred regardless of whether stop-on-match or stop-on-no-match was selected or not. The EOP bit is set only when an EOP ends a DMA transfer; it is not set for EOPs issued during chaining. Note that two or even all three of MC, TC, and EOP may be set if multiple reasons existed for ending the DMA operation. The MCH and MCL bits report on the match state of the upper and lower comparator bytes, respectively. These bits are set when the associated comparator byte has a match; otherwise, they are reset regardless of whether stop-on-match or stop-on-no-match is programmed or not. Regardless of the DMA operation performed, these bits reflect the comparator status at the end of the DMA operation. These two bits are provided to help determine which byte matched or did not match when using 8-bit matches with word searches and transfer-and-searches. The two reserved bits return zeros during reads.

Each channel has an interrupt save register which can be read by the local processor. When an interrupt occurs and IP equals one, either because a DMA operation terminated or because an NXM abort occurred during chaining, the contents of the interrupt vector register and part of the channel status register are stored in the 16-bit interrupt save register shown in Figure 5-17. Because the vector and status are stored, a new vector can be loaded during chaining and a new DMA operation can be performed before an interrupt acknowledge cycle occurs. If another interrupt occurs on the channel before the first is acknowledged, further channel activity is suspended. As soon as the first clear IP command is issued, the status and vector for the second interrupt is loaded into the interrupt save register and channel operation resumes. The controller can retain only two interrupts for each channel; a third operation cannot be initiated until the first interrupt has been cleared.

NOTE

For bits IS01 and IS00 in Figure 5-17, the X indicates that the T-11 will not use these bits as part of the vector address. However, they can be read and written from the vector register and may serve as identifiers to the vector address.

5.5.6 Special-Purpose Registers

The pattern and mask registers are used in search and transfer-and-search operations. Both the pattern and mask registers may be loaded by chaining or may be read or written by the T-11. The pattern register contains the pattern that the read data is compared with. Setting a mask register bit to a one specifies that the bit always matches.

The interrupt vector register (Figure 5-18) contains the vector or identifier to be output during an interrupt acknowledge cycle. The DTC interrupts the local processor on level 4 with an external interrupt. The local processor will only respond during a vector read transaction of an interrupt acknowledge cycle to bits V7-2. Bits V0 and V1, however, may be used as a TAG field since the T-11 can read and write these bits under program or chain control. The vector register is copied into the interrupt save queue when an interrupt occurs.

ADDRESS: 174530, 174532

IV15	IV14	IV13	IV12	IV11	IV10	IV09	IV08	IV07	IV06	IV05	IV04	IV03	IV02	IV01	IV00
X	X	X	X	X	X	X	X	V7	V6	V5	V4	V3	V2	V1	V0

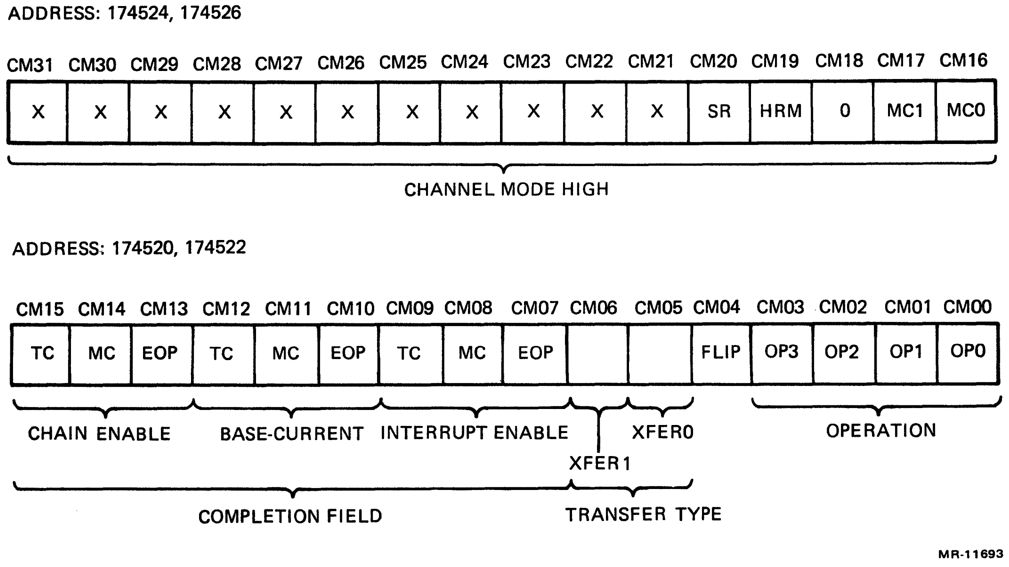
MR-11692

Figure 5-18 Interrupt Vector Register

NOTE

The X bits in Figure 5-18 are disregarded by the controller chip. Bits V1 and V0 are not part of a valid vector address since the processor cannot accept them during an interrupt acknowledge.

Associated with each channel are two channel mode registers (Figure 5-19). There are 21 bits defined in each channel mode register. These bits are described in the following paragraphs. The other 11 bits are unused and are read as zeros. The channel mode register selects what type of DMA operation the channel is to perform, how the operation is to be executed, and what action, if any, is to be taken when the channel finishes.



ADDRESS: 174524, 174526

CM31	CM30	CM29	CM28	CM27	CM26	CM25	CM24	CM23	CM22	CM21	CM20	CM19	CM18	CM17	CM16
X	X	X	X	X	X	X	X	X	X	X	SR	HRM	0	MC1	MC0

CHANNEL MODE HIGH

ADDRESS: 174520, 174522

CM15	CM14	CM13	CM12	CM11	CM10	CM09	CM08	CM07	CM06	CM05	CM04	CM03	CM02	CM01	CM00
TC	MC	EOP	TC	MC	EOP	TC	MC	EOP			FLIP	OP3	OP2	OP1	OP0

MR-11693

Figure 5-19 Channel Mode Register

Operation Field (OP3–0) – These four bits select the type of operation the channel is to perform. It also selects the operand size of words. The code definition for these bits are listed in Table 5-18.

Flip – The flip bit is used to select whether the current ARA register points to the source and the ARB register points to the destination or vice-versa. When cleared to zero, ARA = source and ARB = destination; when set to one, ARA = destination and ARB = source.

Table 5-18 DTC Transfers

OP3	OP2	OP1	OP0	Operation*	Operand Size ARA	ARB	Transaction Type
0	0	0	0	Transfer	Word	Word	Flowthrough
0	0	0	1	Transfer	Byte	Byte	Flowthrough
0	0	1	0	Illegal	–	–	–
0	0	1	1	Illegal	–	–	–
0	1	0	0	Transfer-and-search	Word	Word	Flowthrough
0	1	0	1	Transfer-and-search	Byte	Byte	Flowthrough
0	1	1	0	Illegal	–	–	–
0	1	1	1	Illegal	–	–	–
1	0	0	0	Transfer	Byte	Word	Flowthrough
1	0	0	1	Illegal	–	–	–
1	0	1	0	Illegal	–	–	–
1	0	1	1	Illegal	–	–	–
1	1	0	0	Transfer-and-search	Byte	Word	Flowthrough
1	1	0	1	Illegal	–	–	–
1	1	1	0	Search	Word	Word	Read
1	1	1	1	Search	Byte	Byte	Read

* Byte operations are only supported locally; Q-Bus byte operations will result in word operations being performed.

XFER1, XFER0 – This field selects the type of transfer to be performed by the channel according to the following bit definitions.

XFER1	XFER0	Type of Transfer
0	0	Single transfer
0	1	Demand dedicated with bus hold (software hog mode)
1	0	Demand dedicated with bus release (software hog mode)
1	1	Channel-to-channel demand interleave

The combination of these transfer type bits with software control allows the controller to perform transfers in single operation steps, n operation steps, or bus interleaved operation steps.

Single transfer operation is intended for use with peripherals which transfer single bytes or words at irregular intervals. Each application of a software request command will cause the channel to perform a single iteration of the DMA operation. Each application of a hardware request input will also cause a single iteration of the DMA operation.

Demand dedicated with bus hold operation allows a software request to run to completion. The hardware request causes the DTC to perform iterations as long as the request is asserted and there is a valid operation count.

Negating the request line with a valid operation count causes the DTC to stall while holding the bus. Thus, the user can start or stop execution of DMA operations by modulating the request pin. The demand dedicated with bus release operation allows the DTC to give the bus back to the T-11 when the hardware request negates.

Demand interleave operation forces the DTC to share the bus between its channels; if the master-mode bit MM2 (CPU interleave) is set, the bus is shared with the T-11.

Completion Field – This field is divided into three subfields called chain enable, base-to-current reload enable, and interrupt enable. These fields are used to program the action taken by the channel at the end of a DMA operation. Setting a 1 in any of the bits in this field selects the completion operation to be performed. The reason for ending is stored in the completion status field of the channel's status register. This information is retained until the next DMA operation ends, at which time the status register is updated to reflect the reason(s) for the latest termination. If no actions are selected for the type of termination that occurred, the NAC bit in the status register will be set. More than one action can occur when a DMA operation ends. This may happen if more than one action was programmed for the applicable termination. The interrupt enable completion field for a channel directs a response to an event to be detected by the IP bit in the channel status register. If the CIE channel status bit is set, then a completion field event may set the IP bit and generate an interrupt to the T-11. For any completion field event to set the IP bit, the particular event must be enabled in the completion field. Hence, polled operation can be done by setting channel mode bits 7 and 9 and polling the IP bit in the channel status register with CIE cleared. When the IP bit is tested asserted, then the channel status register contains the valid termination control bits.

MC1, MC0 – This field selects the type of match control operation that is in effect during DMA transfers according to the following bit definitions.

MC1	MC0	Operation
0	0	Stop on no match
0	1	Stop on no match
1	0	Stop on word match
1	1	Stop on byte match

0 – This bit must always be written to zero to enable the channel DMA acknowledge strobe to assert during transfers.

HRM – This bit, when set, masks the channel hardware request line from initiating DMA activity, and may be treated as a don't-care bit since no sources of hardware requests exist to the DTC.

SR – This bit, when set by a processor write or by a chain reload, causes the channel to request the bus and perform transfers according to the XFER 1, XFER 0 operation.

5.6 ASYNCHRONOUS SERIAL LINE UNIT (SLU1)

One asynchronous serial line unit is provided by a custom MOS 40-pin DL UART chip that has data buffers and control and status registers as part of the chip. Serial line unit 1 (SLU1) can be a system console with proper software handling. The assigned register addresses are shown Table 5-19 and the interrupt vector addresses are shown in Table 3-1. The data buffers and the status registers can be accessed by either the T-11 or the DMA controller. The serial data leads have an electrical interface which is compatible with RS422/RS232-C/RS423. The user can optionally add a DLV11-KA EIA to 20 mA converter; however, a reader run strobe is not provided, nor is a 110 baud rate input capability.

Table 5-19 Local I/O Register Address

Address Input	Device Register
177561–177560	SLU1 receive CSR
177563–177562	SLU1 receive buffer
177565–177564	SLU1 transmit CSR
177567–177566	SLU1 transmit buffer

5.6.1 Operating Parameters Selection

Programmable or jumper-selectable baud rates are supported by four XCSR bits which control the baud rate selection (see Paragraph 6.3.11). These baud rates are level sensitive, not latched. This requires the software control of the XCSR. use bit set and bit reset type instructions after the baud rate has been written into the SLU1 driver. The SLU1 driver powers up with a default programmable baud rate of 300. XCSR bits 05–03 can be read or written to define a new baud rate if bit 01 (PBRE) of the XCSR is set and the PBRE shunt is removed. Split speed operation of the transmitter and the receiver is not supported. Providing an external baud rate clock into the SLU1 is not supported. Table 6-8 lists the internal (programmable) baud rates. When the PBRE jumper is installed, the baud rate is selected by three other jumpers. See Chapter 6 for jumper configurations. Note that the default power-up baud rate of 300 may be altered by configuring the BRS02–00 bits and by not configuring the PBRE bit.

5.6.2 Console Functions

In this description, *console* is defined as the interface between the system bus and I/O connector J3 on SLU1. The unique features of serial line unit 1 (SLU1) are the 177566–177560 address assignments, vector addresses 60 and 64 assigned to the RBUF and XBUF, and an interrupt request pin which is a hardware interrupt that is asserted when a BREAK is detected. This pin can be jumpered to cause the T-11 to halt trap to the restart address. SLU2 does not have the capability of causing a T-11 restart directly, since it does not have a pin dedicated to BREAK detect functions. Detected BREAK is asserted when the RCVBRK bit is set in the RBUF by the serial-in signal going from a mark to a space and staying in the space condition for 11 bit times after serial reception starts. This RBUF bit is negated when the serial-in signal returns to the mark condition.

The BREAK detect interrupt remains asserted until either the RBUF is read or +5 V is removed. The BREAK restart condition is jumper selectable and if selected KXTCSRB bits 06–04 will read all ones when a BREAK is detected and before an RBUF read occurs. Hence, the response to this interrupt must be a read of the RBUF.

5.6.3 Interrupt Vectors/Priorities

T-11 internal level 4 vectors 60 and 64 are assigned to the receive and transmit interrupts, respectively. The receive has a higher priority than the transmit.

5.6.4 Address Selection

SLU1 is a 16-bit bus-compatible device in the KXT11-C SBC that responds only to even address word or low byte writes, or to even address reads. Table 5-19 lists the locally addressable I/O page register address assignments.

5.6.5 Register Bit Assignments

The register bit assignments for transmit and receive registers are shown in Figures 5-20 through 5-23.

The following are the bit definitions for Figures 5-20 through 5-23.

RO = read only and is always read as zero if not serving one of the following functions:

FR	= Framing
RW	= Read/write
BRK	= Break
RCV	= Receive
RDY	= Ready
ACT	= Active
X	= Transmit
DN	= Done
M	= Maintenance
IE	= Interrupt enable
PB<2–0>	= Programmable baud rate
ERR	= Error select
OR	= Over run
PBE	= Programmable baud rate, select

RCSR

15	14	13	12	11	IU	09	08	07	06	05	04	03	02	01	00
RO	RO	RO	RO	RO RCV ACT	RO	RO	RO	RO RCV	RO RCV DN	RO IE	RO	RO	RO	RO	RO

MR-12245

Figure 5-20 RCSR Register

RBUF

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RO ERR	RO OR ERR	RO FR ERR	RO	RO RCV BRK	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO	RO

RCV DATA BUFFER

MR-12246

Figure 5-21 RBUF Register

XCSR

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
RO	RO	RO	RO	RO	RO	RO	RO	RO X RDY	RW X IE	RW BP2	RW PB1	RW PBO	RW M	RW PBE	RW X BRK

MR-12247

Figure 5-22 XCSR Register

XBUF

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

TRANSMIT DATA BUFFER

MR-12248

Figure 5-23 XBUF Register

5.7 PARALLEL I/O

The parallel I/O and functions are provided by the Z8036 ZCIO counter/timer and parallel I/O (PIO) transceiver chips. The ZCIO chip provides a counter/timer operating at a frequency of 2 MHz or a resolution of 500 ns.

The PIO chip is protected from the connector by high input impedance, passive pull-up or TTL-compatible driver transceivers. These transceivers provide no glitch protection to the 40-pin connector interface. They comply with IEEE 488 electrical standards. The direction and TTL/open collector options are selected by programming the write only PIO I/O buffer control register at (PCR) address 17777140.

The significant features of the PIO chip are: two independent 8-bit, double buffered, bidirectional I/O ports, plus a 4-bit special-purpose I/O port. The I/O ports feature programmable polarity, programmable direction (bit mode), pulse catchers, and programmable passive pull-up outputs; four handshake modes, including 3-wire (like IEEE-488); pattern-recognition logic, programmable as a 16-vector interrupt controller; three independent 16-bit counter/timers with three output duty cycles (pulsed, one-shot, and square wave), and up to four external access lines for each counter (count input, output, gate, and trigger). The counter/timers are programmable as retriggerable or nonretriggerable.

The ZCIO time constant register contains a value of N which is down counted at a rate of 500 ns per N value or at the rate of the externally supplied clock. An external counter clock input must have a minimum clock low time of 230 ns and a minimum clock high time of 230 ns, with a minimum period of 500 ns. The counter/timers can be operated in three modes: pulse output, one-shot, and square-wave. In pulse output mode the output strobes for one clock pulse (500 ns) N + one clock pulses after the trigger or reload. An example is shown in Figure 5-24.

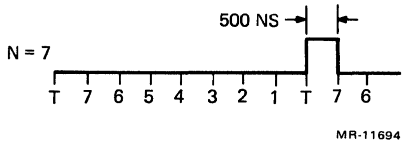


Figure 5-24 Pulse Output Mode Operation

In one-shot mode a time constant of N results in a pulse duration of N clock pulses starting one clock pulse after the trigger. An example is shown in Figure 5-25.

For square wave mode an initial count of N results in a square wave with a period of N clock pulses. For odd values of N, the output is high for (N+1)/2 counts and low for (N−1)/2 counts.

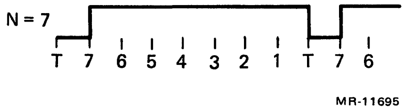


Figure 5-25 One-Shot Mode Operation

5.7.1 Address Selection

The PIO chip is addressable in the local I/O page on even address boundaries from address 17777000 to 17777136. In all PIO transactions, the high byte must be treated as erroneous data. The RJA bit (see Figure 5-26) in the master interrupt control register must be written to zero on power-up or after a reset instruction to ensure that the PIO chip responds to even addresses only. The only accessible register in the PIO chip following power-up or a reset is the master interrupt control register, which must be written with 100 at address 17777000 before any other register can be read or written to.

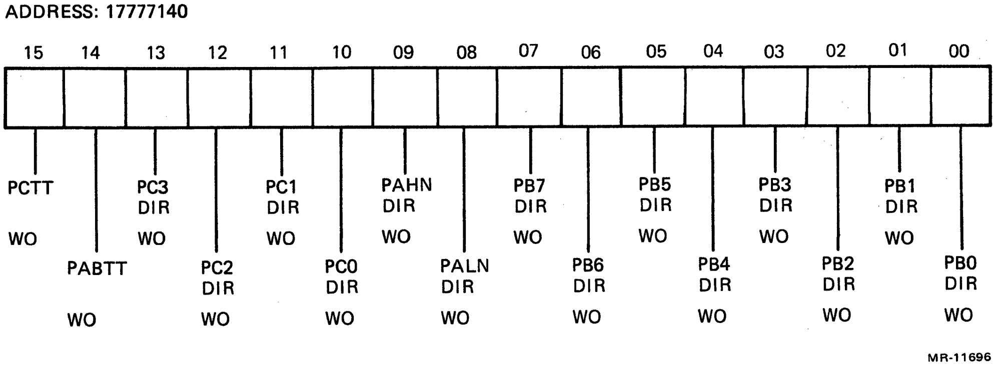


Figure 5-26 I/O Buffer Control Register

5.7.2 Interrupt Vectors/Priorities

The interrupt request is received as a level 5 interrupt and the PIO chip must supply a vector when the interrupt is acknowledged.

The PIO chip must be programmed with three vector addresses, one for each data port and one for the counter and timers. It must be programmed not to provide status with the vector addresses. This is done by writing zeros to the master interrupt control register bits 5, 4, 3, and 2. The vector returned is the contents of the port interrupt vector register or the counter/timer's interrupt vector register. The PC1 request strobe can be configured to operate with the DTC channel A or channel B request input. The wait mode is not supported.

5.7.3 Buffers and I/O Connector

The PIO lines are buffered by bidirectional, high-current buffers from J4, a 40-pin, right-angle shrouded header. The buffers are capable of providing high impedance inputs, with 600 mV of hysteresis and either passive internal pull-up driver outputs or TTL-compatible driver outputs. These buffers comply with IEEE 488 electrical standards. Paragraph 6.7.2 describes the pin assignments of the PIO connector as seen from the SBC edge and provides a signal cross-reference list.

The IEEE 488 buffers restrict to some degree the configurability of the output/input lines. Port A buffers are configurable as nibbles as either input or output or bidirectionally controlled via the PIO I/O buffer control register bits PAHN and PALN. Port B is configurable as an 8-bit port, two 4-bit nibbles, or as a bit port with the direction of each bit controlled via the PIO I/O buffer control register. The four lines of port C can be individually configured to be an input or output as controlled by the PIO buffer control register at address 17777140.

5.7.4 Register Assignments

Table 5-20 lists the PIO registers by address and by name. Refer to the Zilog *Z8036 Counter/Timer and Parallel I/O Unit Manual* for additional details.

Table 5-20 Parallel Port Registers

Address	Name
17777000	Master interrupt control
17777002	Master configuration control
17777004	Port A interrupt vector
17777006	Port B interrupt vector
17777010	Counter/timer interrupt vector
17777012	Port C data path polarity
17777014	Port C data direction
17777016	Port C special I/O control
17777020	Port A command and status
17777022	Port B command and status
17777024	Counter/timer 1's control
17777026	Counter/timer 2's control
17777030	Counter/timer 3's control
17777032	Port A data
17777034	Port B data
17777036	Port C data
17777040	Counter/timer 1's current count MSB
17777042	Counter/timer 1's current count LSB
17777044	Counter/timer 2's current count MSB
17777046	Counter/timer 2's current count LSB
17777050	Counter/timer 3's current count MSB
17777052	Counter/timer 3's current count LSB
17777054	Counter/timer 1's time constant MSB
17777056	Counter/timer 1's time constant LSB
17777060	Counter/timer 2's time constant MSB
17777062	Counter/timer 2's time constant LSB
17777064	Counter/timer 3's time constant MSB
17777066	Counter/timer 3's time constant LSB
17777070	Counter/timer 1's mode specification
17777072	Counter/timer 2's mode specification
17777074	Counter/timer 3's mode specification
17777076	Current vector
17777100	Port A mode specification
17777102	Port A handshake specification
17777104	Port A data path polarity
17777106	Port A data direction
17777110	Port A special I/O control
17777112	Port A pattern polarity
17777114	Port A pattern transition
17777116	Port A pattern mask
17777120	Port B mode specification
17777122	Port B handshake specification
17777124	Port B data path polarity
17777126	Port B data direction
17777130	Port B special I/O control
17777132	Port B pattern polarity
17777134	Port B pattern transition
17777136	Port B pattern mask
17777140	I/O buffer control register

Figures 5-26 through 5-46 show the bit map for each type of register. Table 5-21 defines the I/O buffer control register bits.

Table 5-21 I/O Buffer Control Register Bit Description

Bit	Name	Status	Function
07–00	PB<7–0> DIR	Write only	These bits are cleared on power-up or by writing a zero to configure the PIO port B buffers for receivers. When written with ones, the eight buffers are configured for driving the I/O connector. All lines are in the receive state on power-up or reset.
08	PALN DIR	Write only	This bit is cleared on power-up or by writing a zero to configure the PIO port A low nibble buffers for receivers. When written with a one, the four buffers are configured for driving the I/O connector. All lines are in the receive state on power-up or reset.
09	PAHN DIR	Write only	This bit is cleared on power-up or by writing a zero to configure the PIO port A high nibble buffers for receivers. When written with a one, the four buffers are configured for driving the I/O connector. All lines are in the receive state on power-up or reset.
13–10	PC<3–0> DIR	Write only	These. bits are cleared on power-up or by writing a zero to configure the PIO port C buffers for receivers. When written with a one, the programmed buffer is configured for driving the I/O connector. All lines are in the receive state on power-up or reset.
14	PABTT	Write only	This bit is cleared on power-up or by wrıting a zero to configure the PIO port A and port B driver buffers for open collector. When written with a one, the port A and port B driver buffers are configured for active pull-up. Cleared on reset.
15	PCTT	Write only	This bit is cleared on power-up or by writing a zero to configure the PIO port C driver buffers for open collector. When written with a one, the port C driver buffers are configured for active pull-up. Cleared on reset.

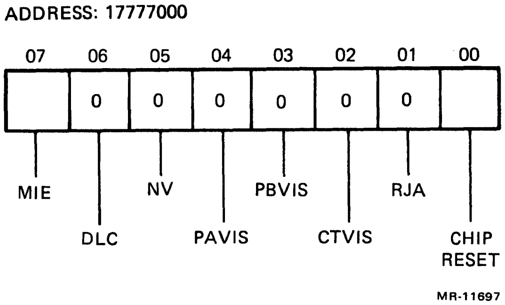
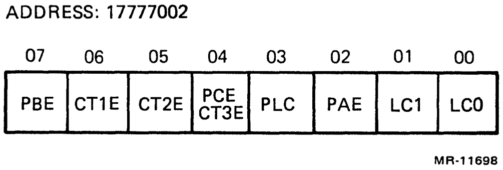


Figure 5-27 Master Interrupt Control Register

- MIE - Master interrupt enable
- DLC - Disable lower chain
- NV - No vector
- PAVIS - Port A vector includes status
- PBVIS - Port B vector includes status
- CTVIS - Counter/timers vector includes status
- RJA - Right justify address
- CHIP RESET


ADDRESS: 17777002

07	06	05	04	03	02	01	00
PBE	CT1E	CT2E	PCE CT3E	PLC	PAE	LC1	LC0

MR-11698

Figure 5-28 Master Configuration Control Register

- PBE - Port B enable
- CT1E - Counter/timer 1 enable
- CT2E - Counter/timer 2 enable
- PCE/CT3E - Port C and counter/timer 3 enable
- PLC - Port link control: 0 = port A/port B independent, 1 = port A/port B linked
- PAE - Port A enable
- LC1, LC0 - Counter/timer link controls:

  00 = counter/timers independent
  01 = C/T1's /output gates C/T2
  10 = C/T1's /output triggers C/T2
  11 = C/T1's /output is C/T2's count input

ADDRESS: 177771000, 17777120

07	06	05	04	03	02	01	00
PTS1	PTS0	ITB	SB	IMO	PMS1	PMS0	LPM DTE

MR-11699

Figure 5-29 Port Mode Specification Register

- PTS1, PTS0 – Port type select:

  00 = bit port
  01 = input port
  10 = output port
  11 = bidirectional port

- ITB – Interrupt on two bytes

- SB – Single buffered mode

- IMO – Interrupt on match only

- PMS1, PMS0 – Pattern mode specification:

  00 = disable pattern match
  01 = AND mode
  10 = OR mode
  11 = OR-priority encoded vector mode

- LPM/DTE – Latch on pattern match (bit mode only)/deskew timer enable (handshake mode)

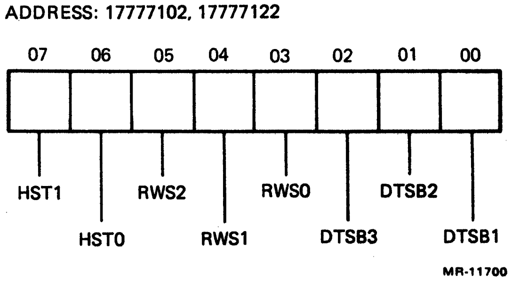


Figure 5-30 Port Handshake Specification Register

- HST1, HST0 – Handshake type specification:

  00 = interlocked handshake
  01 = strobed handshake
  10 = pulsed handshake
  11 = three-wire handshake

- RWS2, RWS1, RWS0 – Request/wait specification:

  000 = disabled
  001 = not supported
  011 = not supported
  100 = special Request
  101 = output Request
  111 = input Request

- DTSB3, DTSB2, DTSB1 – Deskew time specification (specify the 3 MSBs of the deskew timer, LSB is forced 1 and one additional count is added).

  1. Example: 000 = 2 4-MHz clock cycles
  2. Example: 100 = 10 4-MHz clock cycles

ADDRESS: 17777020, 17777022

07	06	05	04	03	02	01	00
IUS	IE	IP	ERR	ORE	IRF	PMF	IOE

MR-11701

Figure 5-31 Port Command and Status Register

- IUS – Interrupt under service
- IE – Interrupt enable
- IP – Interrupt pending

NOTE

Writing IUS, IE, and IP are encoded as follows:

000 null code (do nothing)
001 clear IP and IUS
010 set IUS
011 clear IUS
100 set IP
101 clear IP
110 set IE
111 clear IE

- ERR – Interrupt error read-only
- ORE – Output register empty read-only
- IRF – Input register full read-only
- PMF – Pattern match flag read-only
- IOE – Interrupt on error read/write

ADDRESS: 17777100, 17777120

07	06	05	04	03	02	01	00
DPP7	DPP6	DPP5	DPP4	DPP3	DPP2	DPP1	DPP0

MR-11702

Figure 5-32 Data Path Polarity Register

DPP7–DPP0 – Data path polarity: 0 = noninverting, 1 = inverting

ADDRESS: 17777106, 17777126

07	06	05	04	03	02	01	00
DD7	DD6	DD5	DD4	DD3	DD2	DD1	DD0

MR-11703

Figure 5-33 Data Direction Register

DD7–DD0 – Data direction: 0 = output bit, 1 = input bit

ADDRESS: 17777130

07	06	05	04	03	02	01	00
SIO7	SIO6	SIO5	SIO4	SIO3	SIO2	SIO1	SIO0

MR-11704

Figure 5-34 Special I/O Control Register

SIO7–SIO0 – Special input/output:

- 0 = normal input or output
- 1 = output with open drain (not supported) or input with 1's catcher

ADDRESS: 17777032, 17777034

07	06	05	04	03	02	01	00
PA7 PB7	PA6 PB6	PA5 PB5	PA4 PB4	PA3 PB3	PA2 PB2	PA1 PB1	PA0 PB0

MR-11705

Figure 5-35 Port A, Port B Data Register

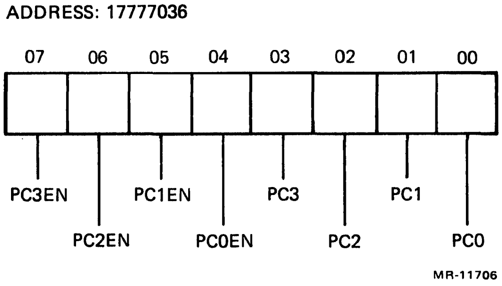


Figure 5-36 Port C Data Register

PC3EN-PC0EN

- 0 = writing of corresponding PC enabled
- 1 = writing of corresponding PC inhibited

ADDRESS: 17777112, 17777132

07	06	05	04	03	02	01	00
PP7	PP6	PP5	PP4	PP3	PP2	PP1	PP0

MR-11707

Figure 5-37 Pattern Polarity Register

ADDRESS: 17777114, 17777134

07	06	05	04	03	02	01	00
PT7	PT6	PT5	PT4	PT3	PT2	PT1	PT0

MR-11708

Figure 5-38 Pattern Transition Register

PM	**PT**	**PP**
0	0	X bit masked off
0	1	X any transition
1	0	0 zero*
1	0	1 one*
1	1	0 one* to zero* logic transition
1	1	1 zero* to one* logic transition

* The definition of zero and one is determined by the data path polarity.

ADDRESS: 17777116, 17777136

07	06	05	04	03	02	01	00
PM7	PM6	PM5	PM4	PM3	PM2	PM1	PM0

MR-11709

Figure 5-39 Pattern Mask Register

ADDRESS: 17777070, 17777072, 17777074

07	06	05	04	03	02	01	00
C/SC	EOE	ECE	ETE	EGE	RE	DSC1	DSC0

MR-11710

Figure 5-40 Counter/Timer Mode Specification Register

- C/SC - Continuous/single cycle 1 = continuous
- EOE - External output enable
- ECE - External count enable
- ETE - External trigger enable
- EGE - External gate enable
- RE - Retrigger enable bit
- DSC1, DSC0 - Output duty cycle select:

  00 = pulse output
  01 = one-shot output
  10 = square-wave output
  11 = illegal

ADDRESS: 17777024, 17777026, 17777030

07	06	05	04	03	02	01	00
IUS	IE	IP	ERR	RCC	GCB	TCB	CIP

MR-11711

Figure 5-41 Counter/Timer Command and Status Register

- IUS - Interrupt under service read/write
- IE - Interrupt enable read/write
- IP - Interrupt pending read/write

NOTE

Writing IUS, IE, and IP are encoded as follows:

000 null code (do nothing)
001 clear IP and IUS
010 set IUS
011 clear IUS
100 set IP
101 clear IP
110 set IE
111 clear IE

- ERR - Interrupt error read-only
- RCC - Read counter control (only cleared by reading CCR LSB)
- GCB - Gate command read/write
- TCB - Trigger command write-only, read returns
- CIP - Count in progress read-only

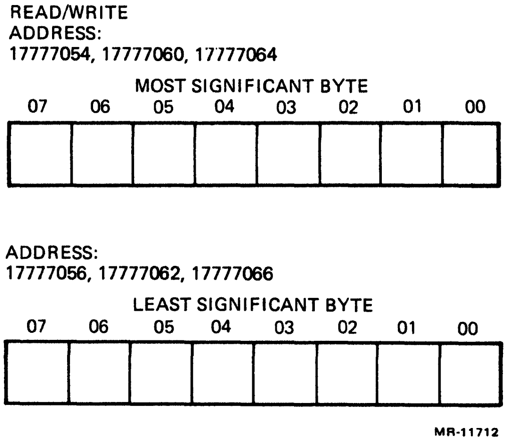


Figure 5-42 Counter/Timer Time Constant Register

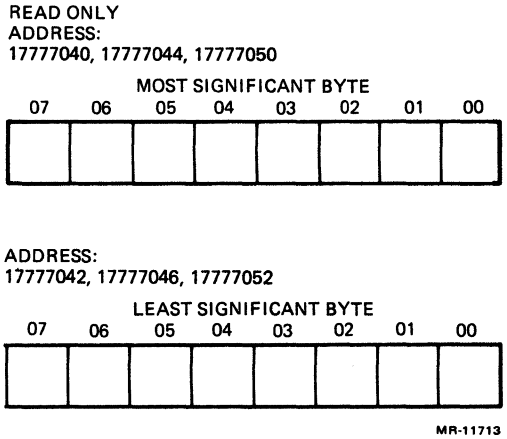


Figure 5-43 Counter/Timer Current Count Register

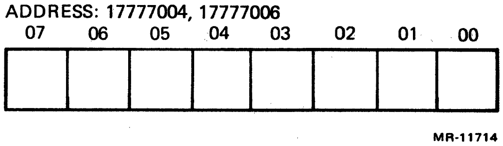


Figure 5-44 Port A and B Interrupt Vector Register

Port vector status: bits 03, 02 and 01 are modified if VIS = 1 or MIE = 1 according to:

- Priority encoded mode = number of highest priority bit with a match
- All other modes = ORE, IRF, PMF (output empty, input full, pattern match flag)
- 000 = error

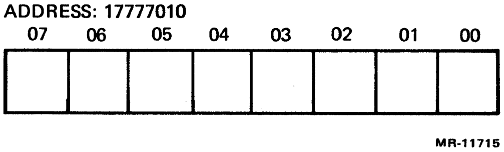


Figure 5-45 Counter/Timer Interrupt Vector Register

Counter/timer status: bits 02 and 01 are modified to indicate interrupting counter according to:

00 = C/T3 source
01 = C/T2 source
10 = C/T1 source
11 = error

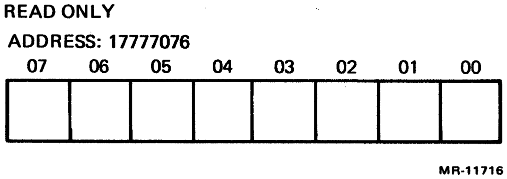


Figure 5-46 Current Vector Register

Interrupt vector based on highest priority unmasked IP. If no IP is present, this register reads all ones.

5.8 SYNCHRONOUS/ASYNCHRONOUS SERIAL COMMUNICATIONS (SLU2)

To facilitate remote synchronous or asynchronous I/O communications, a two-channel, serial I/O port controller is provided. This port controller is a NEC UPD7201 multiprotocol, two-channel, serial communications controller. Channel A complies with RS449 mandatory interchange circuits for a type SR (send-receive) data transmission element. Channel B complies with RS449 mandatory interchange circuits for a type DT (data and timing only) data transmission element. The programmable baud rate generator provides fundamental clocking rates to the controller providing variable programmable data rates of times (×) 1, 16, 32, or 64, thus facilitating data rates from 150 baud to 76.8 K baud. The controller attributes are: double-buffered transmitter data and triple-buffered received data; programmable CRC algorithm; asynchronous operation with 5, 6, 7, or 8 data bits; 1, 1-1/2, 2 stop bits; odd, even, or no parity; break generation and detection; interrupt on parity, overrun, or framing errors; monosync, bisync, and external sync operations; software selectable sync characters; auto-sync insertion; CRC generation and checking; HDLC and SDLC operations; abort sequence generation and detection; auto-zero insertion and detection; address field recognition; CRC generation and checking; and I-field residue handling. Refer to the *UPD7201 Technical Manual* for additional information. The RS449 modem interface is supported by an Intel 8255A-5 chip operated in mode 0, with ports A and C lower outputs, and ports B and C upper inputs. This chip provides A, B and C (KXTCSRA, KXTCSRB, KXTCSRC) low bytes.

5.8.1 Programmable Baud Rate Generation

Programmable baud rates are supplied by an Intel 8254-2 timing controller chip with two counters operated at fundamental frequency 9.8304 MHz for baud rate generation for both channels A and B.

The third counter is supplied with an 800-Hz clock and is a general-purpose counter used to facilitate watchdog timers or software waits. KXTCSRA bit 07 controls a level 6 (vector 104) interrupt request from the counter/timer OUT pin. The 8254-2 has three independent 16-bit counters, all of the GATE inputs are fixed to the one state. The 9.8304-MHz and the 800-Hz clocks can be scaled in BCD or binary. Counters zero and one support the UPD 7201 baud rate clock.

The clock rates that feed the 7201 controller are computed as follows. For the 8254-2 chip mode 3 even counts of n, the clock period is n (9.8304-MHz or 800-Hz) clock periods, and clock high time = clock low time. Odd count values must not be used for baud rate generation. Tables 5-22 and 5-23 provide the decimal divider ratio (n) required by the 8254-2 chip for a specified baud rate. Note that asynchronous baud rate n values have assumed that the UPD 7201 controller divides by 16.

Table 5-22 Actual Synchronous Bit Rate

Synchronous bit rate = 9830.4 K/divider ratio(N)

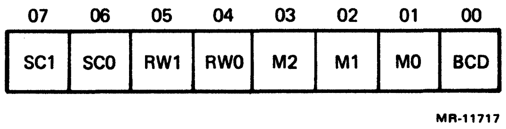
Ratio (Decimal)	Actual Bit Rate	Error (%)
136	72.282 K	.3917
176	55.855 K	.2597
204	47.720 K	.3922
512	19.2 K	
1024	9.6 K	
2048	4.8 K	
8192	1.2 K	

Table 5-23 Actual Asynchronous Baud Rate

Asynchronous baud rate = [9830.4 K(1/16)]/divider ratio

Ratio (Decimal)	Actual 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	1.2 K
1024	600
2048	300
4096	150
5586	109.989

Counter/timer 2 can be used as a general-purpose software counter/timer in mode 0 (interrupt on terminal count), mode 2 (rate generator), mode 3 (square wave mode), or mode 4 (software triggered strobe). When programming the 8254-2 chip, control register is sent a command which indicates a counter and a mode or a counter operation to follow the command. The write-only timer control register at address 175736 has the format shown in Figure 5-47.



07	06	05	04	03	02	01	00
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

Figure 5-47 Timer Control Register

Bit 00 (BCD) – Zero state indicates binary counter 16-bits; one state indicates binary coded decimal (BCD) counter with 4 decades.

Bits 01, 02, 03 (M2, M1, M0) – Mode select bits used to configure the timer modes.

000	Mode 0 – interrupt on terminal count
001	Mode 1 – not supported
010	Mode 2 – rate generator
011	Mode 3 – square wave mode
100	Mode 4 – software triggered strobe
101	Mode 5 – not supported
110	Reserved – don't use
111	Reserved – don't use

Bits 04, 05 (RW1, RW0) – Read/write sequence selection:

00	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

Bits 06, 07 (SC1, SC0) – Select counter for command:

00	Select counter 0
01	Select counter 1
10	Select counter 2
11	Read-back command

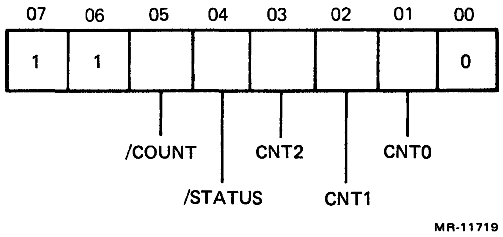
To avoid reading a changing count, the counter latch command should be used. The counter latch command format is shown in Figure 5-48.

	07	06	05	04	03	02	01	00
COUNTER 0	0	0	X	X	X	X	X	X
COUNTER 1	0	1	0	0	X	X	X	X
COUNTER 2	1	0	X	X	X	X	X	X

MR-11718

Figure 5-48 Counter Latch Register

Issuing this command causes the current counter value to be latched and held in a buffer either until it is read or the mode is changed. The read-back command may also be used to obtain count and status information for a particular counter. The read-back command format as shown in Figure 5-49.



Figure˙5-49 ˙ Current Counter Value Register

The selected counter is determined by bits 03, 02, and 01 being set. The command may be used to latch multiple counter output latches by setting the count bit and selecting the desired counter(s). Once a counter is latched, it must be read before another command is valid. The read-back command may also be used to latch status information of a selected counter(s) by setting the status bit. Status must be latched to be read from that counter's data register. The status byte format is shown in Figure 5-50.

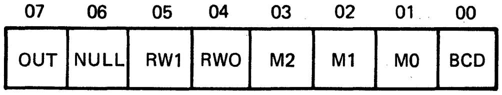


Figure 5-50 Counter Status Register

Bits 05–00 are the same as those issued to a counter through the control register. The null bit is asserted to indicate that a count value may not be loaded into a counter just yet. When the null bit is cleared, the counter is active. Bit 07 will be a one when the OUT pin is not asserted and a zero when the OUT pin is asserted low. Three typical 8254 load sequences follow.

```
MODE 3:                        MODE 3:                        MODE 3:

MOVB    #67, @#175736          MOVB    #226,@#175736          MOVB    #126,@#176736
MOVB    #LSB,@#175730          MOVB    #15, @#175734          MOVB    #100,@#175732
MOVB    #MSB,@#175730
```

5.8.2 Address Selection

The control registers and data buffers for the UPD 7201 controller and the 8254-2 chip are addressable by the local processor at addresses 175736–175700. KXTCSR A register is located at address 177520 and supports some of the SLU2 modem options. KXTCSR B is located at address 177522 and supports the test mode signal for the modem interface. The control registers and data buffers for the SLU2 controller and the 8254-2 chip are divided into read-only and write-only groups. Reading a write-only register address returns indeterminate data but does not alter the state of any register bits. Writing read-only addressable registers does not alter the register contents. These low-byte registers are addressable on even address boundaries only, and for the UPD 7201, and 8254-2, the high must be treated as erroneous. There are 16 addressable registers: eight are SLU2 controller registers and eight are timer registers. Table 5-24 defines the addressable 8254-2 registers and Table 5-25 defines the addressable SLU2 controller registers.

Table 5-24 Timer (8254-2) Addressable Registers

Address	Name	Function
175736	Timer control register	Write only
175734	Timer 2 data register	Write only
175734	Timer 1 data register	Write only
175730	Timer 0 data register	Write only
175726	X valid address	Read only indeterminate data
175724	Timer 2 data register	Read only
175722	Timer 1 data register	Read only
175720	Timer 0 data register	Read only

Table 5-25 SLU2 Controller Addressable Registers

Address	Name	Function
175716	Channel B transmitter	Write only
175714	Channel B control register	Write only
175712	Channel B receiver	Read only
175710	Channel B status	Read only
175706	Channel A transmitter	Write only
175704	Channel A control register	Write only
175702	Channel A receiver	Read only
175700	Channel A status	Read only

5.8.3 Interrupt Vectors/Priorities

The UPD 7201 controller must be operated in nonvectored interrupt mode, or in either of two vectored interrupt modes: in channel A DMA/channel B non-DMA mode, or channel A and B non-DMA mode. Vectored interrupt mode is selected by writing a 025 or a 024 into the channel A control register 2 at address 175704. The receiver then has a higher internal interrupt priority than the transmitter, which has a higher internal priority than the modem interrupts. The vector includes the status bit that must also be set since the 7201 controller interrupts the T-11 on level 4 through internal vector 70. The service routine must be able to determine the cause of the interrupt by reading the interrupt vector from control register 2 channel B and then the IP bit from channel A status register. Timer 2 can interrupt the T-11 on level 6 via the OUT pin through vector 104. This interrupt is maskable via KXTCSR A bit 7 control. When the controller is operated in DMA mode, the request signals to the DTC must be configured by installing jumpers from the controller request pins to the DTC request pins. See Chapter 6 for jumper sections. The wait mode or block mode of operation is not supported.

5.8.4 I/O Drivers and Receivers

The electrical interface complies with RS449 standards, which implies that RS422 and RS232-C interfacing is available. In addition, channel B may be operated in party line mode by controlling the party line transmitter through UPD transmitter control register at address 175714 (register pointer 5). The receiver is enabled via KXTCSR A bit 1 and is enabled when the bit is in the zero state. Note that all modem control signals asserted from the SLU2 controller or KXTCSR A are asserted by writing a one. All received control or status signals are read asserted in the one state.

5.8.5 Register Bit Assignments (CR0 through CR7; SR0 through SR2)

The asynchronous/synchronous UPD controller chip register bit assignments are detailed in the following paragraphs.

COMMAND REGISTER (CHANNEL A) (See Figure 5-51.)

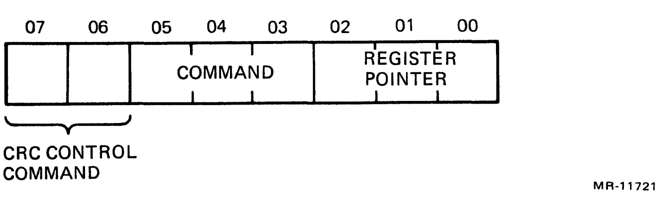


Figure 5-51 Command Control and Pointer Control Register (CR0)

Register Pointer
The register pointer specifies the next accessed register number. After a hardware/software reset, this field is set to 0. The first control byte always goes to control register 0. When this field is set to a value other than zero, the next control or status access is to the specified register.

Commands
000 Null – No effect, used when setting a CRC command.

001 Send Abort – SDLC mode abort code.

010 Reset External/Status Interrupts – Clears a pending interrupt and reenables the latches so that new external/status changes are detected.

011 Channel Reset – Resets the internal interrupt prioritization logic. Must wait one NOP instruction before writing a new command.

100 Enable Interrupt on Next Character – When in interrupt on first received character mode, this command reenables the interrupt logic for the next received character.

101 Reset Pending Transmitter Interrupt /DMA Request – Resets a pending transmitter buffer becoming empty interrupt or DMA request without sending another character.

110 Error Reset – Resets a special receive condition interrupt. It also reenables the parity and overrun error latches that allow you to check for these errors at the end of a message.

111 End Of Interrupt – Unblocks lower priority interrupt requests, must be issued at some point in the interrupt service routine.

CRC Control Commands
00 Null – No effect, used when issuing other commands.

01 Reset Receiver CRC Checker – Resets the CRC checker to zero for synchronous mode and resets to all ones when in SDLC mode.

10 Reset Transmitter CRC Generator – Resets the CRC generator to zero for synchronous mode and resets to all ones when in SDLC mode.

11 Reset Idle/CRC Latch L – Resets the idle/CRC latch so that when a transmitter underrun condition occurs, the transmitter enters the CRC phase of operation and begins to send the 16-bit CRC character calculated up to that point. The latch is then set so that if the underrun condition persists, idle characters are sent following the CRC. After a hardware or software reset, the latch is in the set state.

INTERRUPT CONTROL REGISTER (CHANNEL A) (See Figure 5-52.)

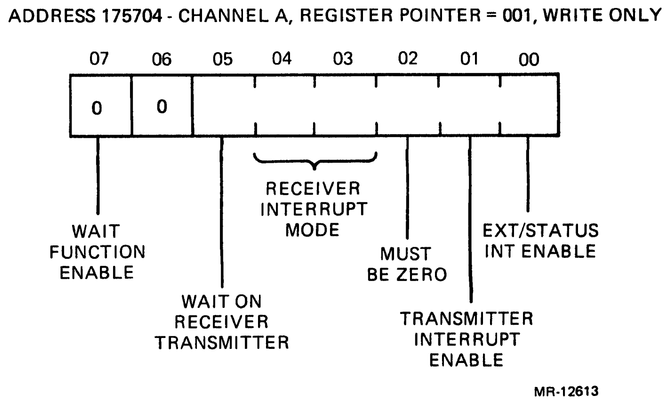

ADDRESS 175704 - CHANNEL A, REGISTER POINTER = 001, WRITE ONLY
07
06
05
04
03
02
01
00
0
0
WAIT FUNCTION ENABLE
WAIT ON RECEIVER TRANSMITTER
RECEIVER INTERRUPT MODE
MUST BE ZERO
TRANSMITTER INTERRUPT ENABLE
EXT/STATUS INT ENABLE
MR-12613

Figure 5-52 Interrupt Control Register (CR1)

Receiver Interrupt Mode
00 Receiver Interrupt/DMA Request Disabled – Polled operation mode.

01 Interrupt on first received character only and issue DMA Request DMA Mode – Interrupt is issued only for the first character received after an enable interrupt on first character command.

10 Interrupt and issue a DMA request if DMA mode on all received characters – An interrupt is issued whenever there is a character present in the receiver buffer (parity error is a special receive condition).

11 Interrupt and issue a DMA request if DMA mode on all received characters – Same as 10 except that parity error is not considered a special receive condition. The vector status will reflect the special error conditions: receiver overrun, asynchronous framing error, parity error, SDLC end of message.

PROCESSOR/BUS INTERFACE REGISTER (CHANNEL A) (See Figure 5-53.)

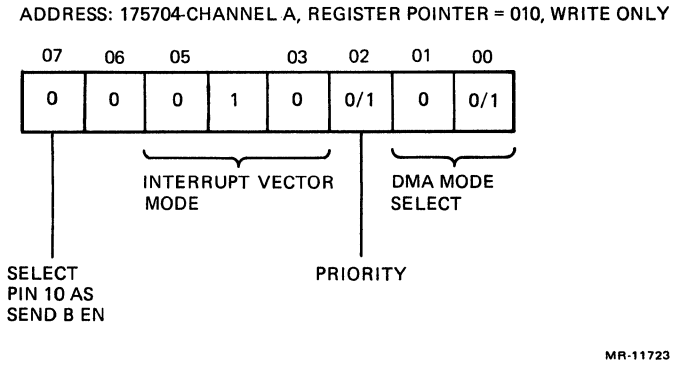

ADDRESS: 175704-CHANNEL A, REGISTER POINTER = 010, WRITE ONLY
07
06
05
03
02
01
00
0
0
0
1
0
0/1
0
0/1
SELECT PIN 10 AS SEND B EN
INTERRUPT VECTOR MODE
PRIORITY
DMA MODE SELECT
MR-11723

Figure 5-53 Processor/Bus Interface Control Register (CR2)

DMA Mode Select	DMA Priority	Interrupt Priority
00 Non-DMA	–	0 RxA>TxA>RxB>TxB>ExTA>ExTB
	–	1 RxA>RxB>TxA>TxB>ExTA>ExTB
01 DMA channel A,	RxA>TxA	0 RxA>RxB>TxB>ExTA>ExTB
Non-DMA channel B	RxA>TxA	1 RxA>RxB>TxB>ExTA>ExTB
10 Not supported		
11 Illegal		

RECEIVER CONTROL REGISTER (CHANNEL A) (See Figure 5-54.)

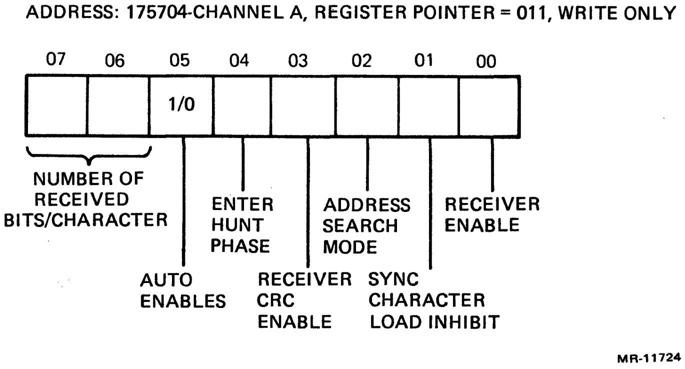

ADDRESS: 175704-CHANNEL A, REGISTER POINTER = 011, WRITE ONLY
07
06
05
04
03
02
01
00
1/0
NUMBER OF RECEIVED BITS/CHARACTER
ENTER HUNT PHASE
ADDRESS SEARCH MODE
RECEIVER ENABLE
AUTO ENABLES
RECEIVER CRC ENABLE
SYNC CHARACTER LOAD INHIBIT
MR-11724

Figure 5-54 Receiver Control Register (CR3)

Number of Received Bits/Character

00	5
01	7
10	6
11	8

MODE CONTROL REGISTER (CHANNEL A) (See Figure 5-55.)

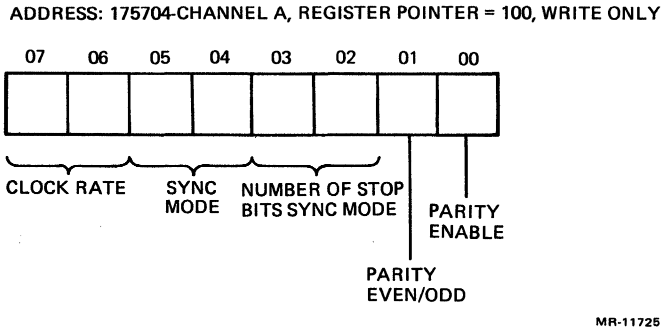

ADDRESS: 175704-CHANNEL A, REGISTER POINTER = 100, WRITE ONLY
07
06
05
04
03
02
01
00
CLOCK RATE
SYNC MODE
NUMBER OF STOP BITS SYNC MODE
PARITY ENABLE
PARITY EVEN/ODD
MR-11725

Figure 5-55 Mode Control Register (CR4)

Number of Stop Bits/Sync Mode

00 Synchronous modes
01 Asynchronous 1 bit time (1 stop bit)
10 Asynchronous 1-1/2 bit times (1 1/2 stop bits)
11 Asynchronous 2 bit times (2 stop bits)

Sync Mode

00 8-bit internal synchronization character (monosync)
01 16-bit internal synchronization character (bisync)
10 SDLC
11 Not supported

Clock Rate

00 Clock rate = 1 × data rate
01 Clock rate = 16 × data rate
10 Clock Rate = 32 × Data Rate
11 Clock Rate = 64 × Data Rate

TRANSMITTER CONTROL REGISTER (CHANNEL B) (See Figure 5-56.)

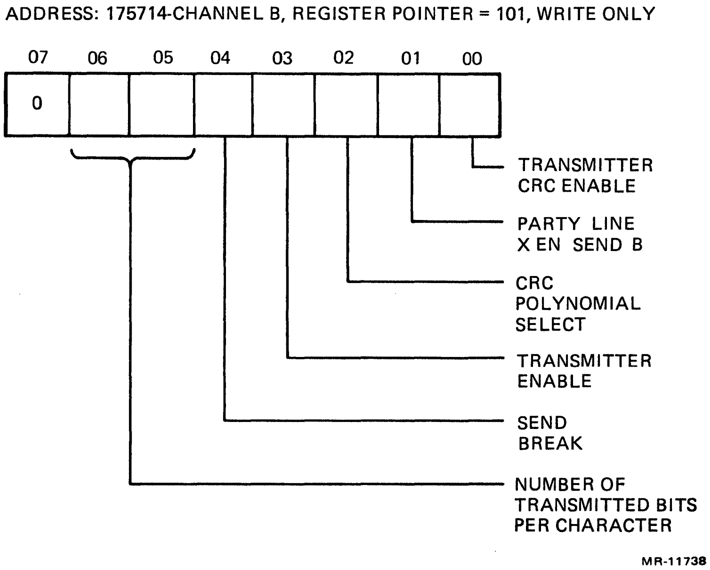


Figure 5-56 Transmitter Control Register (CR5)

Number of Transmitted Bits/Character

00	5 or less
01	7
10	6
11	8

NOTE

For five or less data bits/character, The data must be formatted as follows:

1111000d	**1 bit/character**
111000dd	**2 bits/character**
11000ddd	**3 bits/character**
1000dddd	**4 bits/character**
000ddddd	**5 bits/character**

SYNC/ADDRESS CHARACTER REGISTER (CHANNEL A) (See Figure 5-57.)

SYNC CHARACTER REGISTER (See Figure 5-58.)

STATUS REGISTER (CHANNEL A) (See Figure 5-59.)

STATUS REGISTER 1 (CHANNEL A) (See Figures 5-60, 5-61, and 5-62.)

INTERRUPT STATUS VECTOR REGISTER (CHANNEL B ONLY) (See Figure 5-63.)

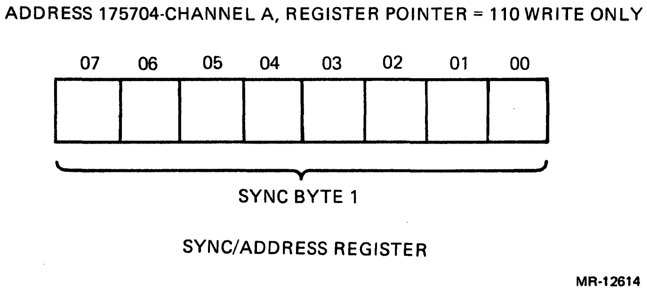


Figure 5-57 Sync/Address Character Register (CR6)

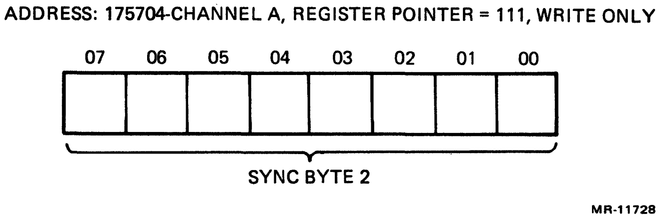

ADDRESS: 175704-CHANNEL A, REGISTER POINTER = 111, WRITE ONLY
07
06
05
04
03
02
01
00
SYNC BYTE 2
MR-11728

Figure 5-58 Sync Character Register (CR7)

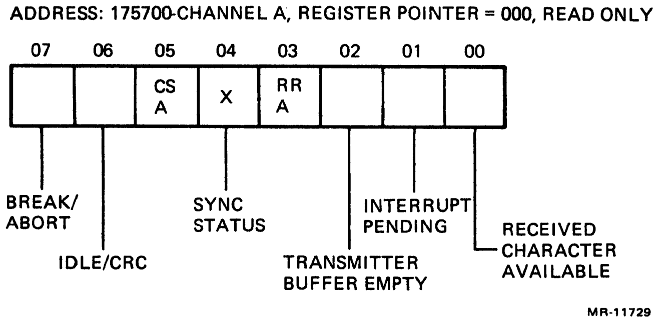


Figure 5-59 Buffer and (External/Status) Status Register (SR0)

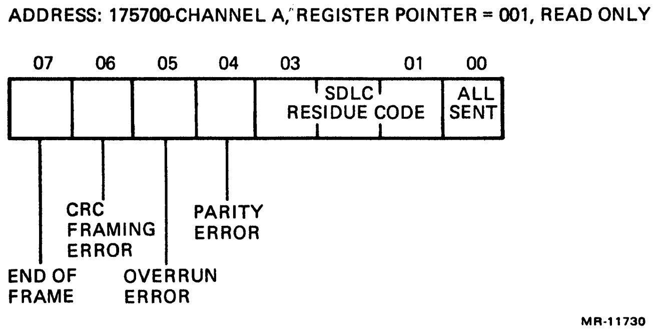


Figure 5-60 Received Character Error and Special Condition Status Register (SR1)

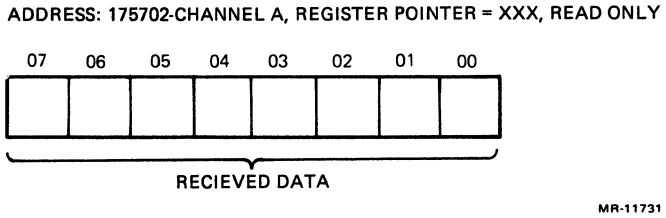

ADDRESS: 175702-CHANNEL A, REGISTER POINTER = XXX, READ ONLY
07
06
05
04
03
02
01
00
RECIEVED DATA
MR-11731

Figure 5-61 Receiver Data Buffer Register

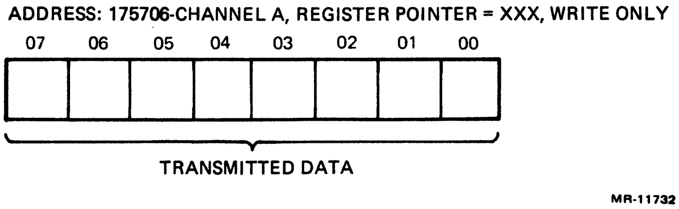

ADDRESS: 175706-CHANNEL A, REGISTER POINTER = XXX, WRITE ONLY
07
06
05
04
03
02
01
00
TRANSMITTED DATA
MR-11732

Figure 5-62 Transmit Data Buffer Register

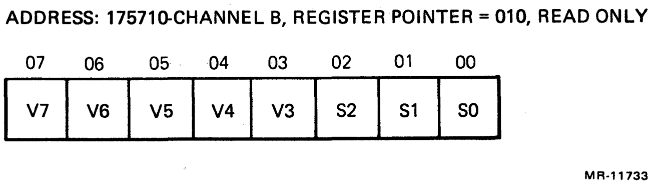

ADDRESS: 175710-CHANNEL B, REGISTER POINTER = 010, READ ONLY
07
06
05
04
03
02
01
00
V7
V6
V5
V4
V3
S2
S1
S0
MR-11733

Figure 5-63 Interrupt Vector Status Register (SR2)

Since the 7201 controller is operated with condition affects vector mode enabled, the value of the vector written to the register at 175714 register pointer = 010 is modified as follows.

S2	S1	S0	Field Description
1	1	1	No interrupt pending
0	0	0	Channel B transmitter buffer empty
0	0	1	Channel B external/status change
0	1	0	Channel B received character available
0	1	1	Channel B special receive condition
1	0	0	Channel A transmitter buffer empty
1	0	1	Channel A external/status change
1	1	0	Channel A received character available
1	1	1	Channel A special receive condition

Status 111 can mean either channel A special receive condition or no interrupt pending. To determine which condition is in effect, examine the interrupt pending bit of status register 0, channel A. Note that in nonvectored interrupt mode you must read the vector register first for interrupt pending to be valid.

COMMAND CONTROL AND POINTER CONTROL REGISTER (See Figure 5-64.)

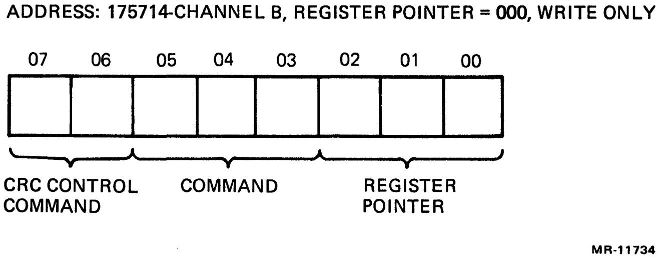


Figure 5-64 Command Control and Pointer Control Register (CR0)

Commands

000 Null – No effect, used when setting a CRC command.

001 Send Abort – SDLC mode abort code.

010 Reset External/Status Interrupts – Clears a pending interrupt and reenables the latches so that new external/status changes are detected.

011 Channel Reset – Resets the internal interrupt prioritization logic. Must wait one NOP instruction before writing a new command.

100 Enable Interrupt on Next Character – When in interrupt on first received character mode, this command reenables the interrupt logic for the next received character.

101 Reset Pending Transmitter Interrupt/DMA Request – Resets a pending transmitter buffer becoming empty interrupt or DMA request without sending another character.

110 Error Reset – Resets a special receive condition interrupt. it also reenables the parity and overrun error latches that allow you to check for these errors at the end of a message.

111 End Of Interrupt – Unblocks lower priority interrupt requests; must be issued at some point in the interrupt service routine.

CRC Control Commands
00 Null – No effect, used when issuing o there commands.

01 Reset Receiver CRC Checker – Resets the CRC checker to zero for synchronous mode and resets to all ones when in SDLC mode.

10 Reset Transmitter CRC Generator – Resets the CRC generator to zero for synchronous mode and resets to all ones when in SDLC mode.

11 Reset Idle/CRC Latch – Resets the idle/CRC latch so that when a transmitter underrun condition occurs the transmitter enters the CRC phase of operation and begins to send the 16-bit CRC character calculated up to that point. The latch is then set so that if the underrun condition persists, idle characters are sent following the CRC. After a hardware or software reset, the latch is in the set state.

INTERRUPT CONTROL REGISTER (See Figure 5-65.)

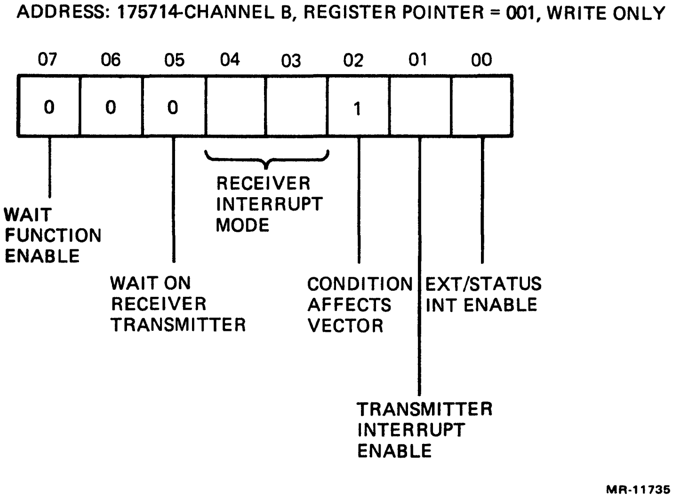

ADDRESS: 175714-CHANNEL B, REGISTER POINTER = 001, WRITE ONLY
07
06
05
04
03
02
01
00
0
0
0
1
WAIT FUNCTION ENABLE
RECEIVER INTERRUPT MODE
WAIT ON RECEIVER TRANSMITTER
CONDITION AFFECTS VECTOR
EXT/STATUS INT ENABLE
TRANSMITTER INTERRUPT ENABLE
MR-11735

Figure 5-65 Interrupt Control Register (CR1)

Receiver Interrupt Mode
00 Receiver Interrupt/DMA Request Disabled – Polled operation mode.

01 Interrupt on first received character only and issue DMA request if DMA mode – Interrupt is issued only for the first character received after an enable interrupt on first character command.

10 Interrupt and issue a DMA request if DMA mode on all received characters – An interrupt is issued whenever there is a character present in the receiver buffer (parity error is a special receive condition).

11 Interrupt and issue a DMA request if DMA mode on all received characters – Same as 10 except that parity error is not considered a special receive condition. The vector status will reflect the special error conditions: receiver overrun, asynchronous framing error, parity error, or SDLC end of message.

Condition Affects Vector

0 Fixed vector interrupt acknowledge sequence is initiated.

1 Modified vector, which reflects condition causing interrupt; interrupt acknowledge sequence is initiated.

RECEIVER CONTROL REGISTER (See Figure 5-66.)

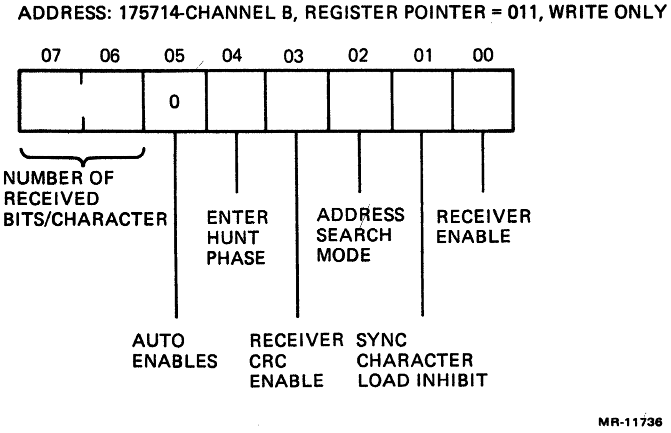

ADDRESS: 175714-CHANNEL B, REGISTER POINTER = 011, WRITE ONLY
07
06
05
04
03
02
01
00
0
NUMBER OF RECEIVED BITS/CHARACTER
ENTER HUNT PHASE
ADDRESS SEARCH MODE
RECEIVER ENABLE
AUTO ENABLES
RECEIVER CRC ENABLE
SYNC CHARACTER LOAD INHIBIT
MR-11736

Figure 5-66 Receiver Control Register

Number of Received Bits/Character

00	5
01	7
10	6
11	8

Auto Enables
Setting this bit allows receiver ready and clear to send modem lines to enable the receiver.

MODE CONTROL REGISTER (See Figure 5-67.)

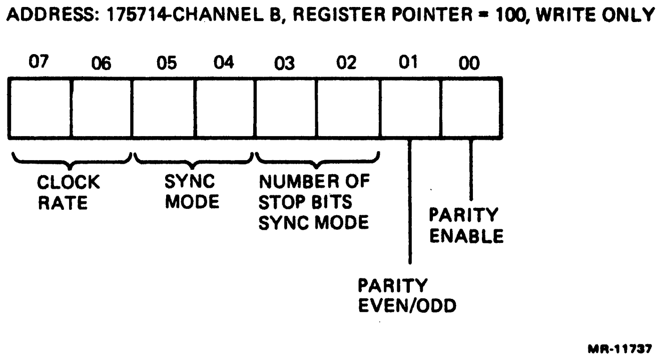

ADDRESS: 175714-CHANNEL B, REGISTER POINTER = 100, WRITE ONLY
07
06
05
04
03
02
01
00
CLOCK RATE
SYNC MODE
NUMBER OF STOP BITS SYNC MODE
PARITY ENABLE
PARITY EVEN/ODD
MR-11737

Figure 5-67 Mode Control Register

Number of Stop Bits/Sync Mode

00 Synchronous modes
01 Asynchronous 1 bit time (1 stop bit)
10 Asynchronous 1 1/2 bit times (1 1/2 stop bits)
11 Asynchronous 2 bit times (2 stop bits)

Sync Mode

00 8-bit internal synchronization character (monosync)
01 16-bit internal synchronization character (bisync)
10 SDLC
11 Not supported

Clock Rate

00 Clock rate = 1 × data rate
01 Clock rate = 16 × data rate
10 Clock rate = 32 × data rate
11 Clock rate = 64 × data rate

TRANSMITTER CONTROL REGISTER (See Figure 5-68.)

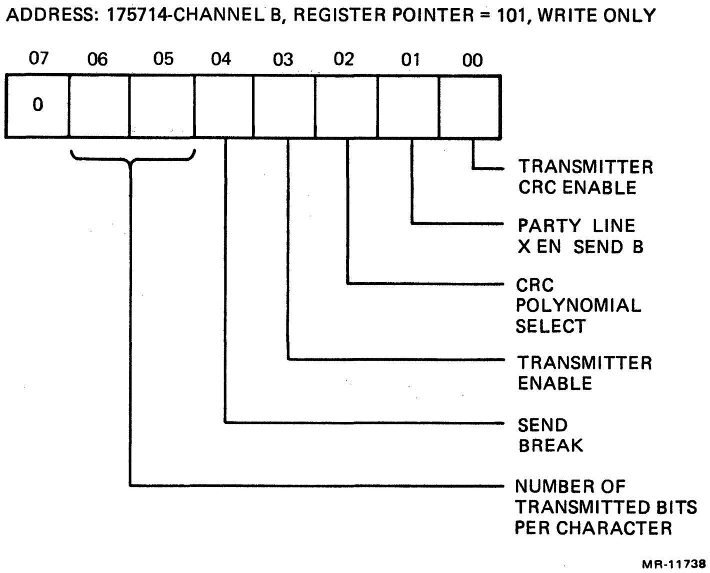


Figure 5-68 Transmitter Control Register

Number of Transmitted Bits/Character

00	5 or less
01	7
10	6
11	8

NOTE

For 5 or less data bits/character, the data must be formatted as follows:

1111000D	**1 bit/character**
111000DD	**2 bits/character**
11000DDD	**3 bits/character**
1000DDDD	**4 bits/character**
000DDDDD	**5 bits/character**

SYNC/ADDRESS CHARACTER REGISTER (See Figure 5-69.)

SYNC CHARACTER REGISTER (See Figure 5-70.)

BUFFER AND EXTERNAL/STATUS REGISTER (See Figure 5-71.)

RECEIVED CHARACTER ERROR AND SPECIAL CONDITION STATUS REGISTER (See Figure 5-72.)

RECEIVER/TRANSMITTER DATA BUFFER REGISTERS (See Figure 5-73 and 5-74.)

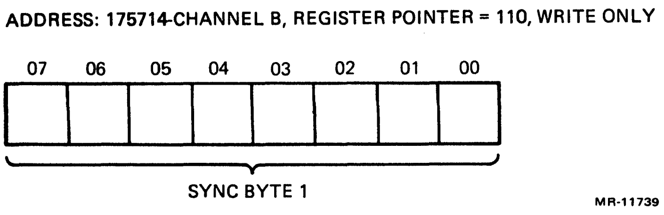

ADDRESS: 175714-CHANNEL B, REGISTER POINTER = 110, WRITE ONLY
07
06
05
04
03
02
01
00
SYNC BYTE 1
MR-11739

Figure 5-69 Sync/Address Character Register

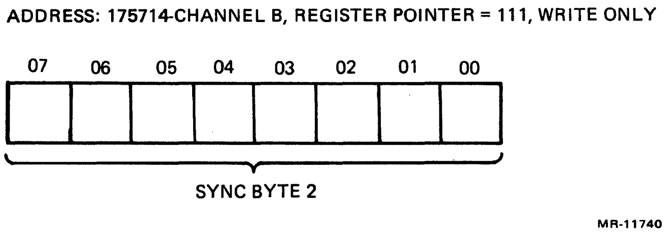

ADDRESS: 175714-CHANNEL B, REGISTER POINTER = 111, WRITE ONLY
07
06
05
04
03
02
01
00
SYNC BYTE 2
MR-11740

Figure 5-70 Sync Character Register

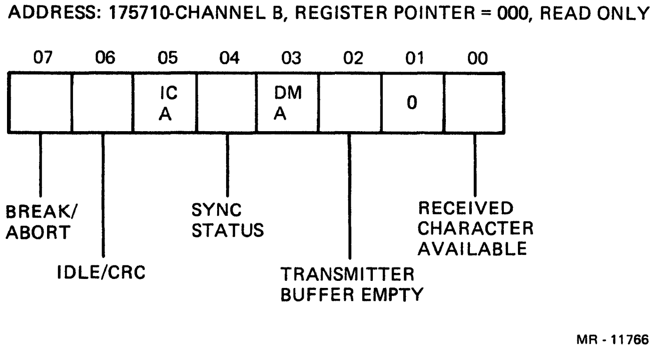


Figure 5-71 Buffer and (External Status) Status Register

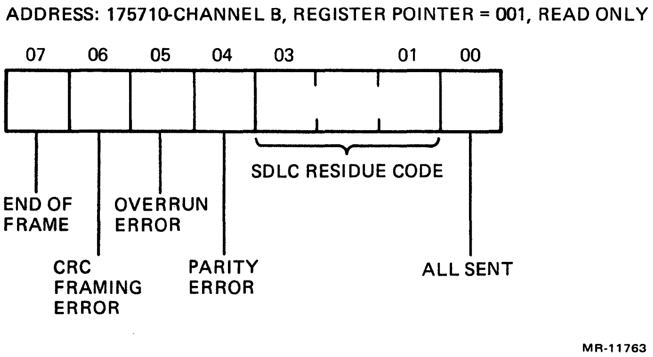


Figure 5-72 Received Character Error and Special Condition Status Register

ADDRESS: 175712-CHANNEL B, REGISTER POINTER = XXX, READ ONLY

<table>
<tr><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="8">RECEIVED DATA</td></tr>
</table>

MR-11764

Figure 5-73 Receiver Data Buffer Register

ADDRESS: 175716-CHANNEL B, REGISTER POINTER = XXX, WRITE ONLY

<table>
<tr><th>07</th><th>06</th><th>05</th><th>04</th><th>03</th><th>02</th><th>01</th><th>00</th></tr>
<tr><td colspan="8">TRANSMITTED DATA</td></tr>
</table>

MR-11765

Figure 5-74 Transmiter Data Buffer Register

CHAPTER 6
CONFIGURATION/INSTALLATION/VERIFICATION

6.1 INTRODUCTION

The configuration, installation and verification of the KXT11-CA single-board computer is described in this chapter. The following four steps, which are an integral part of the procedures, are covered in detail.

NOTE
Leave the factory configuration as is until KXT11-CA SBC performance is verified.

1. Installing jumpers to select operational features
2. Selecting and connecting an appropriate power supply
3. Providing appropriate cables to connect external devices to the serial line and parallel I/O interfaces
4. Verifying operation of the SBC

6.2 CONFIGURATION/INSTALLATION/VERIFICATION PROCEDURE

The configuration/installation/verification procedure shown in Figure 6-1 should be completed by the user to ensure proper SBC operation for the application selected.

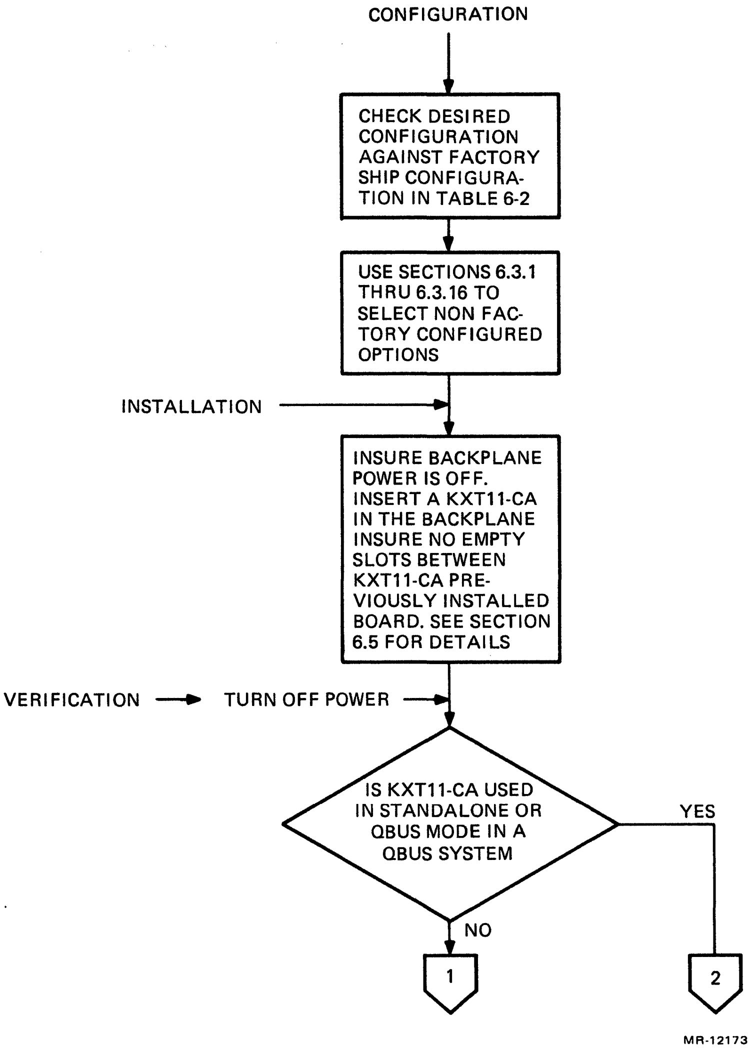

CONFIGURATION
CHECK DESIRED CONFIGURATION AGAINST FACTORY SHIP CONFIGURATION IN TABLE 6-2
USE SECTIONS 6.3.1 THRU 6.3.16 TO SELECT NON FACTORY CONFIGURED OPTIONS
INSTALLATION
INSURE BACKPLANE POWER IS OFF. INSERT A KXT11-CA IN THE BACKPLANE INSURE NO EMPTY SLOTS BETWEEN KXT11-CA PREVIOUSLY INSTALLED BOARD. SEE SECTION 6.5 FOR DETAILS
VERIFICATION
TURN OFF POWER
IS KXT11-CA USED IN STANDALONE OR QBUS MODE IN A QBUS SYSTEM
YES
NO
1
2
MR-12173

Figure 6-1 KXT11-CA SBC Configuration/Installation/Verification Procedure Flow Diagram (Sheet 1 of 3)

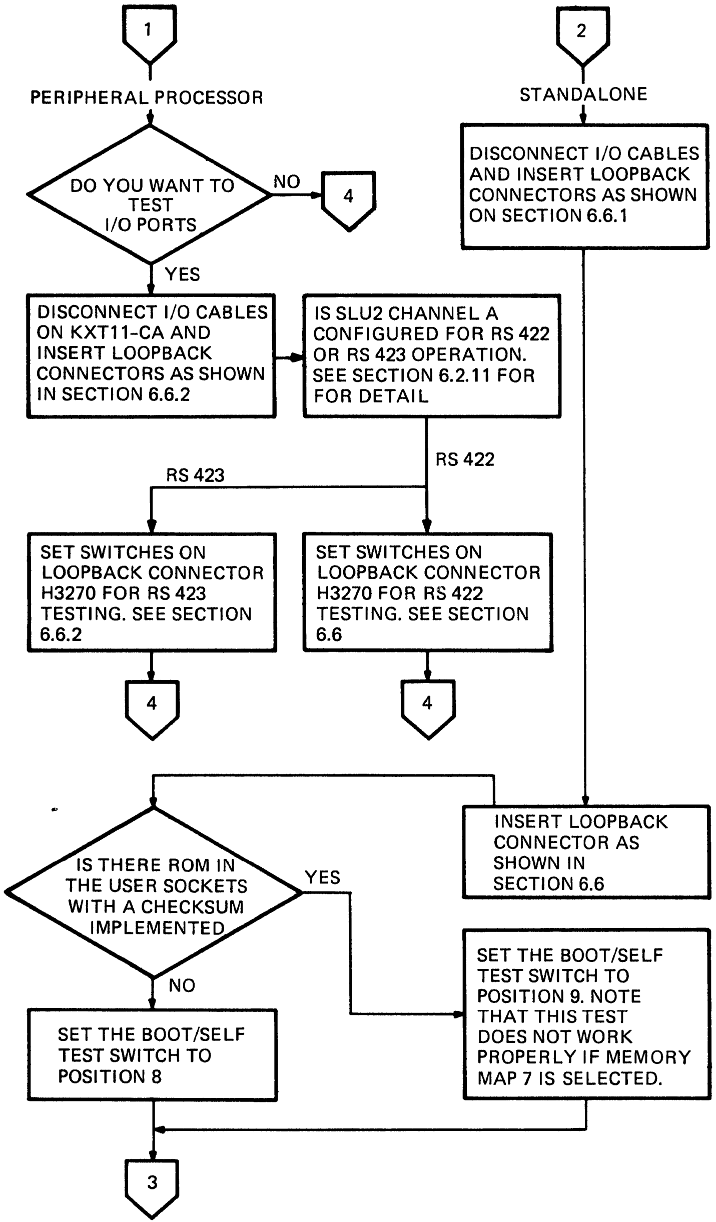

1
PERIPHERAL PROCESSOR
DO YOU WANT TO TEST I/O PORTS
NO
4
YES
DISCONNECT I/O CABLES ON KXT11-CA AND INSERT LOOPBACK CONNECTORS AS SHOWN IN SECTION 6.6.2
IS SLU2 CHANNEL A CONFIGURED FOR RS 422 OR RS 423 OPERATION. SEE SECTION 6.2.11 FOR FOR DETAIL
RS 423
RS 422
SET SWITCHES ON LOOPBACK CONNECTOR H3270 FOR RS 423 TESTING. SEE SECTION 6.6.2
SET SWITCHES ON LOOPBACK CONNECTOR H3270 FOR RS 422 TESTING. SEE SECTION 6.6
4
4
2
STANDALONE
DISCONNECT I/O CABLES AND INSERT LOOPBACK CONNECTORS AS SHOWN ON SECTION 6.6.1
INSERT LOOPBACK CONNECTOR AS SHOWN IN SECTION 6.6
IS THERE ROM IN THE USER SOCKETS WITH A CHECKSUM IMPLEMENTED
YES
NO
SET THE BOOT/SELF TEST SWITCH TO POSITION 8
SET THE BOOT/SELF TEST SWITCH TO POSITION 9. NOTE THAT THIS TEST DOES NOT WORK PROPERLY IF MEMORY MAP 7 IS SELECTED.
3
MR-12174

Figure 6-1 KXT11-CA SBC Configuration/Installation/Verification Procedure Flow Diagram (Sheet 2 of 3)

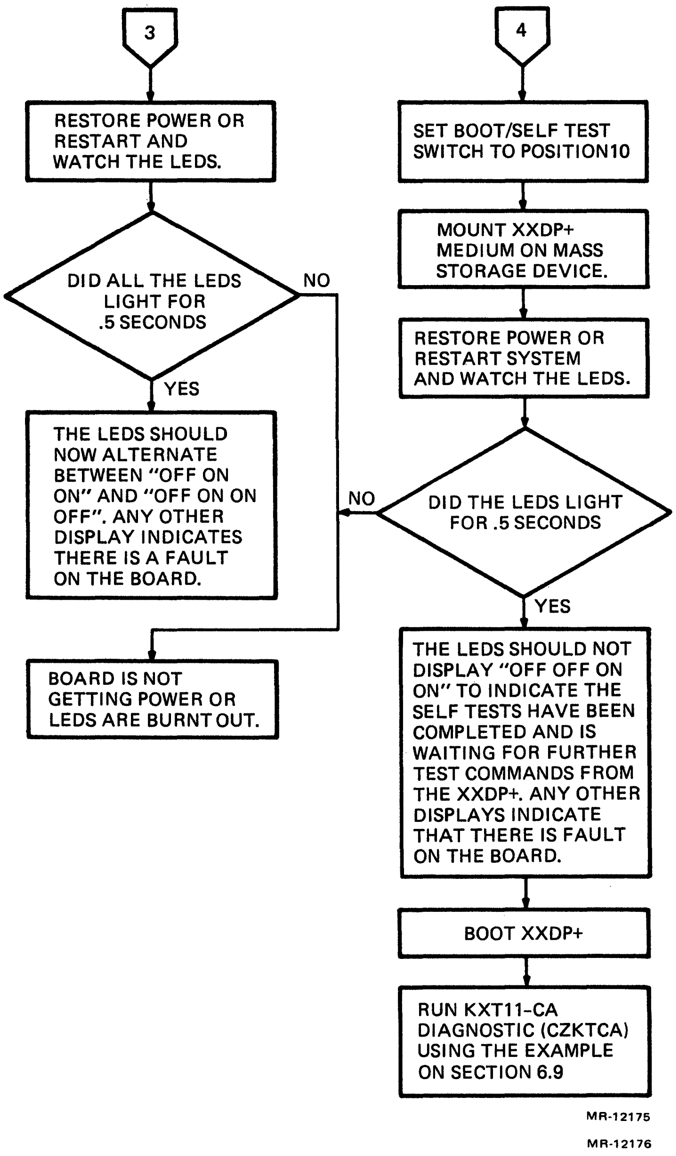
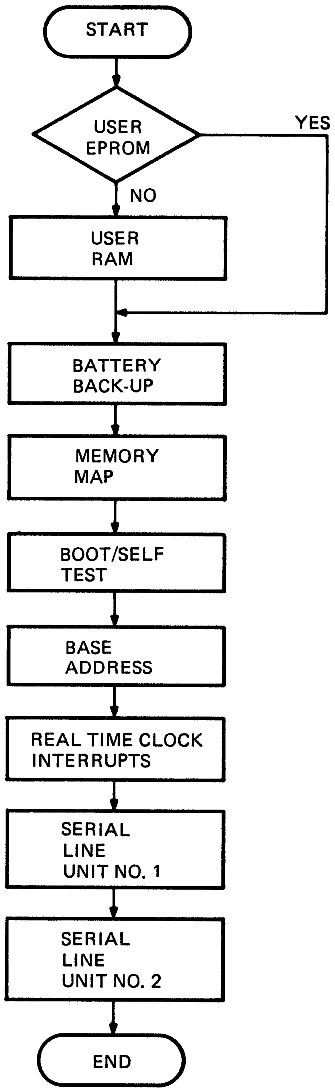

3
RESTORE POWER OR RESTART AND WATCH THE LEDS.
DID ALL THE LEDS LIGHT FOR .5 SECONDS
NO
YES
THE LEDS SHOULD NOW ALTERNATE BETWEEN "OFF ON ON" AND "OFF ON ON OFF". ANY OTHER DISPLAY INDICATES THERE IS A FAULT ON THE BOARD.
BOARD IS NOT GETTING POWER OR LEDS ARE BURNT OUT.
4
SET BOOT/SELF TEST SWITCH TO POSITION10
MOUNT XXDP+ MEDIUM ON MASS STORAGE DEVICE.
RESTORE POWER OR RESTART SYSTEM AND WATCH THE LEDS.
DID THE LEDS LIGHT FOR .5 SECONDS
NO
YES
THE LEDS SHOULD NOT DISPLAY "OFF OFF ON ON" TO INDICATE THE SELF TESTS HAVE BEEN COMPLETED AND IS WAITING FOR FURTHER TEST COMMANDS FROM THE XXDP+. ANY OTHER DISPLAYS INDICATE THAT THERE IS FAULT ON THE BOARD.
BOOT XXDP+
RUN KXT11-CA DIAGNOSTIC (CZKTCA) USING THE EXAMPLE ON SECTION 6.9
MR-12175
MR-12176

Figure 6-1 KXT11-CA SBC Configuration/Installation/Verification Procedure Flow Diagram (Sheet 3 of 3)

6.3 SELECTING OPERATING FEATURES

The KXT11-CA SBC has 32 jumperable functions and two-edge mounted switches which you use to configure the SBC for operating modes necessary to meet your requirements. (See Figure 6-2.) Jumper configurations are done by either installing, repositioning, or removing jumpers between pins. The location and identification number of the pins are shown in Figure 6-3. Table 6-1 defines the pins by functional groups. You can select the operating features you need by connecting the jumper pins with the 29 jumpers that are supplied with the board. The standard factory shipped jumper arrangement is shown in Figure 6-3 and is listed in Table 6-2.



MR-12177

Figure 6-2 KXT11-CA SBC Operating Features Flow Diagram

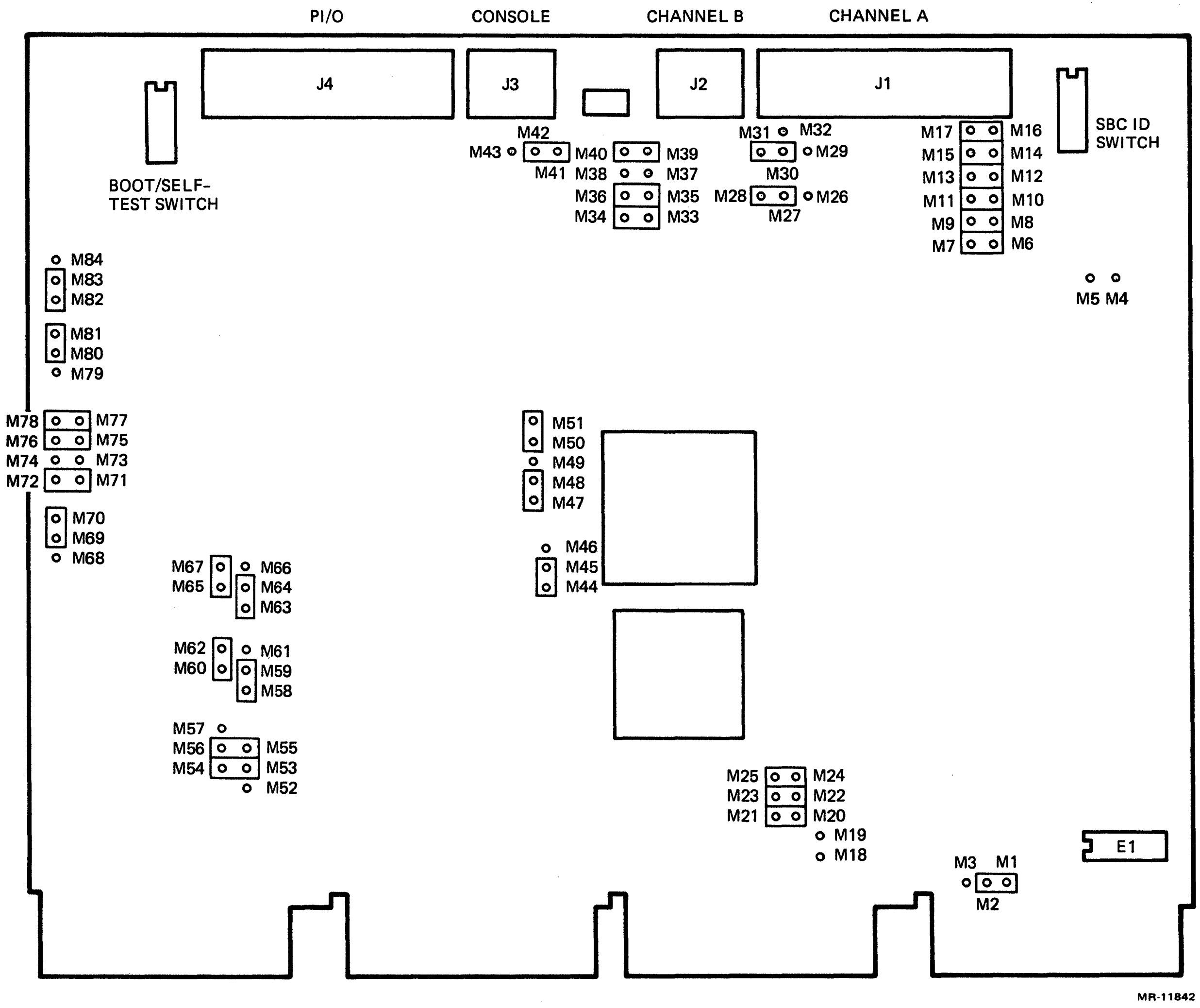


Figure 6-3 KXT11-CA SBC Jumper Layout

Table 6-1 Functional Jumper Pin Definition

Function	Jumpers
Base Address	M18 and M19
DMA Requests	M47, M48, M49, M50, M59
Parallel port	M49
Real-Time clock (60-Hz or 50-Hz)	M84, M82, M83
SLU1 break (nonmaskable)	M81, M80
Memory	
User socket sites	M57, M66, M61, M55, M60, M65, M63, M58, M52, M67, M64, M59, M56, M53, M62
Memory map selection	M24, M25, M22, M23, M20, M21
Battery backup	
On-Board RAM User RAM (if installed)	M1, M2, M3 M68, M69, M70
SLU1	
Programmable baud rate	M72, M74, M76, M78, M75, M73, M71, M77
Serial output (RS422, RS423)	M43, M41, M42
Serial input	M5, M4
SLU2	
Channel A	M13, M12, M17, M16, M11, M10, M9, M8, M15, M14, M7, M6
Channel B	M39, M40, M38, M37, M46, M45, M44, M34, M33, M28, M26, M35, M36, M29, M32, M27, M31, M30, M43

Table 6-2 Factory Shipped Jumper Configuration

Function	Jumpers Installed*
Base address	Low range (jumper out)
Interrupts	
SLU2 channel A	M51 to M50
SLU2 channel B	M47 to M48
Real-Time clock RTC60	M82 to M83
SLU1 (break)	M81 to M80
Memories	
2 K × 8 EPROM	M53 to M55
Memory map 0	No jumpers installed
Battery backup (no)	M1 to M2
User RAM (no)	M69 to M70
SLU1	
Serial baud/bits	No jumpers installed
Rate (300)	
Receiver RS422	M4 to M5
SLU2	
Channel A receiver	
RS422	M15 to M14 M13 to M12 M11 to M10 M7 to M6 M8 to M9
Channel B receiver	
RS422 option	M17 to M16 M37 to M38 M33 to M34 M35 to M30
Channel B transmitter	
RS422 option	M28 to M27 M32 to M30 M39 to M40
Boot/Selftest switch	Position 10

* There are no connections to the following jumper pins: M3, M18, M19, M20, M21, M22, M23, M24, M25, M26, M29, M31, M41, M42, M43, M44, M45, M46, M49, M52, M54, M56, M57 through M68, M71 through M79, and M84.

Paragraphs 6.3.1 through 6.3.16 contain specific jumper configuration information for each functional grouping of jumpers. You can use this information to select the correct configuration for your application. Each paragraph contains a brief description about the particular functional grouping. The SBC layout drawing follows, with an expanded view of the jumper grouping that is being described. Following the layout drawing are all the possible jumper configurations for that particular grouping and a description of each. An asterisk to the left of jumpers denotes the factory shipped configuration.

6.3.1 User EPROM, EEPROM and RAM

The user can install EPROMs, EEPROMs or RAMs into the two user socket sites (Figures 6-4 and 6-5). Tables 6-3 and 6-4 list jumper configurations for several memory chips. The user may select chips from other vendors, however, pin configurations must be compatible. Selected EPROMs or RAMs must meet a maximum enable time of 250 ns and a maximum access time of 450 ns.

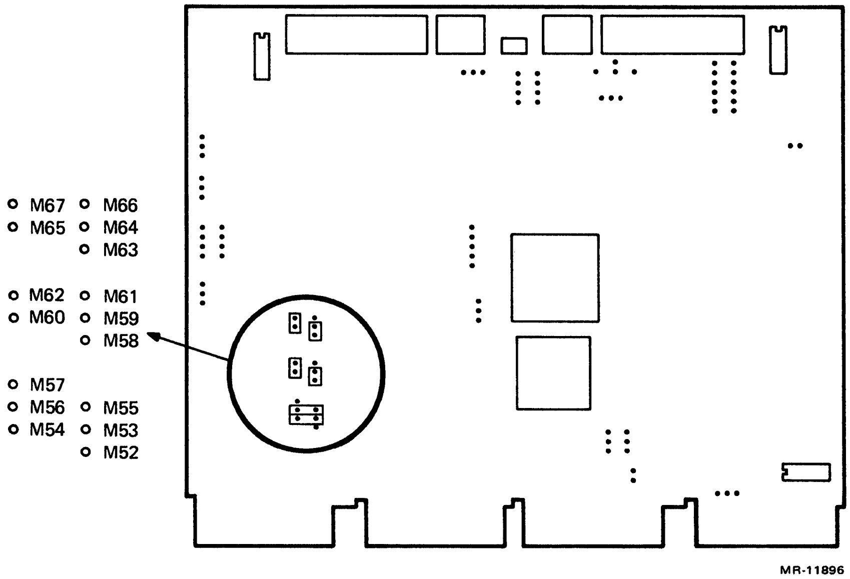


Figure 6-4 User EPROM and RAM Jumper Configurations

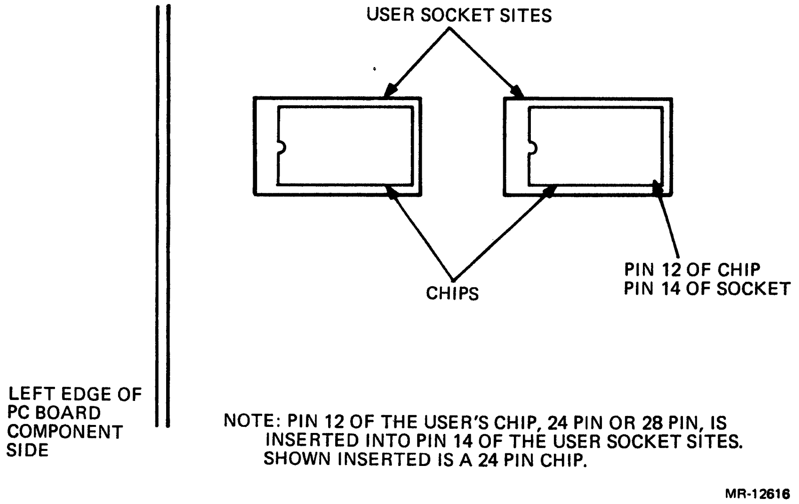

USER SOCKET SITES
CHIPS
PIN 12 OF CHIP
PIN 14 OF SOCKET
LEFT EDGE OF
PC BOARD
COMPONENT
SIDE
NOTE: PIN 12 OF THE USER'S CHIP, 24 PIN OR 28 PIN, IS
INSERTED INTO PIN 14 OF THE USER SOCKET SITES.
SHOWN INSERTED IS A 24 PIN CHIP.
MR-12616

Figure 6-5 User Socket Sites

Table 6-3 EPROM Configuration

Intel Part No.	Pins	Size	Pin Number								
			56	61	55	52	61	58	66	66	63
			to	to	to	to	to	to	to	to	to
			55	62	53	53	59	59*	67	64	64*
2716	24†	2 K × 8	Out	Out	In	Out	In	Out	Out	In	Out
2732	24†	4 K × 8	Out	Out	In	Out	Out	In	Out	Out	In
2764	28	8 K × 8	In	In	Out	Out	Out	In	In	Out	In
27128	28	16 K × 8	In	In	Out	In	Out	In	In	Out	In

Table 6-4 RAM Configurations

Toshiba Part No.	Pins	Size	Pin Number								
			62*	53*	53	59	67*	64	59*	64*	56
			to	to	to	to	to	to	to	to	to
			60	54	55	60	65	65	58	63	55
TMM2016P-01	24†	2 K × 8	Out	Out	In	In	Out	In	Out	Out	Out
TMM5565P-1	28‡	8 K × 8	In	In	Out	Out	In	Out	In	In	In

* Factory shipped configuration.

† For proper insertion of 24-pin chips into the user sockets, see Figure 6-5.

‡ When 28-pin RAM devices are installed, M57 is jumpered to either M56 (+5 V source) or M55 (+5 V battery backup source).

6.3.2 Battery Backup

The user can select the battery backup configuration to maintain a +5 Vdc battery supply to the 16 K static RAM after the dc power is removed. (See Figure 6-6.)

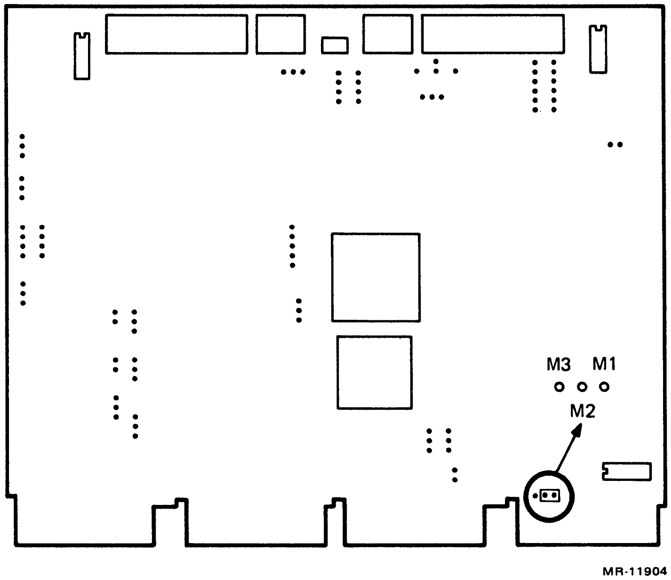


Figure 6-6 Battery Backup Selection

Jumper Connection	Description
*M3 M2 M1 $o \ \boxed{o \quad o}$	No battery backup for RAMs.
M3 M2 M1 $\boxed{o \quad o} \ o$	Battery backup for RAMs selected. The user must supply battery backup voltage to pin AV1.

*Factory shipped configuration.

6.3.3 User Socket +5 V Power

If the 32 Kb static RAM is jumpered for battery backup, then the user socket also can be jumpered for battery backup. Refer to Figure 6-7.

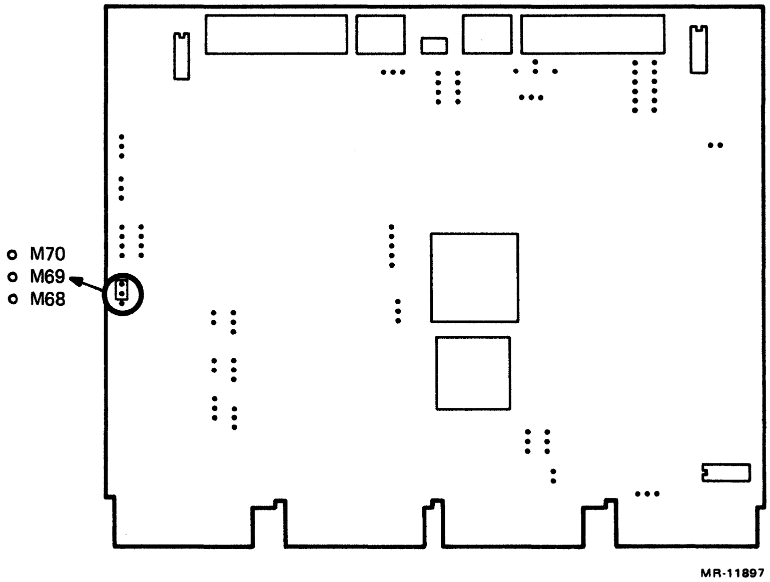


Figure 6-7 User Socket +5 V Power

Jumper Connection	Description
o M70 o M69* o M68	Jumpers +5 V from the Q-Bus source.
o M70 o M69 o M68	Jumpers +5 V from the on-board 32 Kb static RAM +5 V source.

*Factory shipped configuration.

6.3.4 Memory Maps

Figure 6-8 shows and Table 6-5 lists the jumper configuraton for the eight memory maps (Figure 6-9) available to the user.

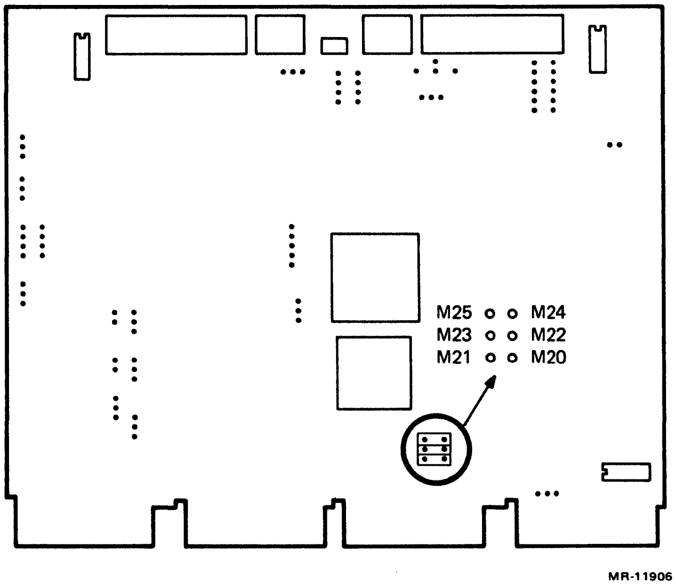
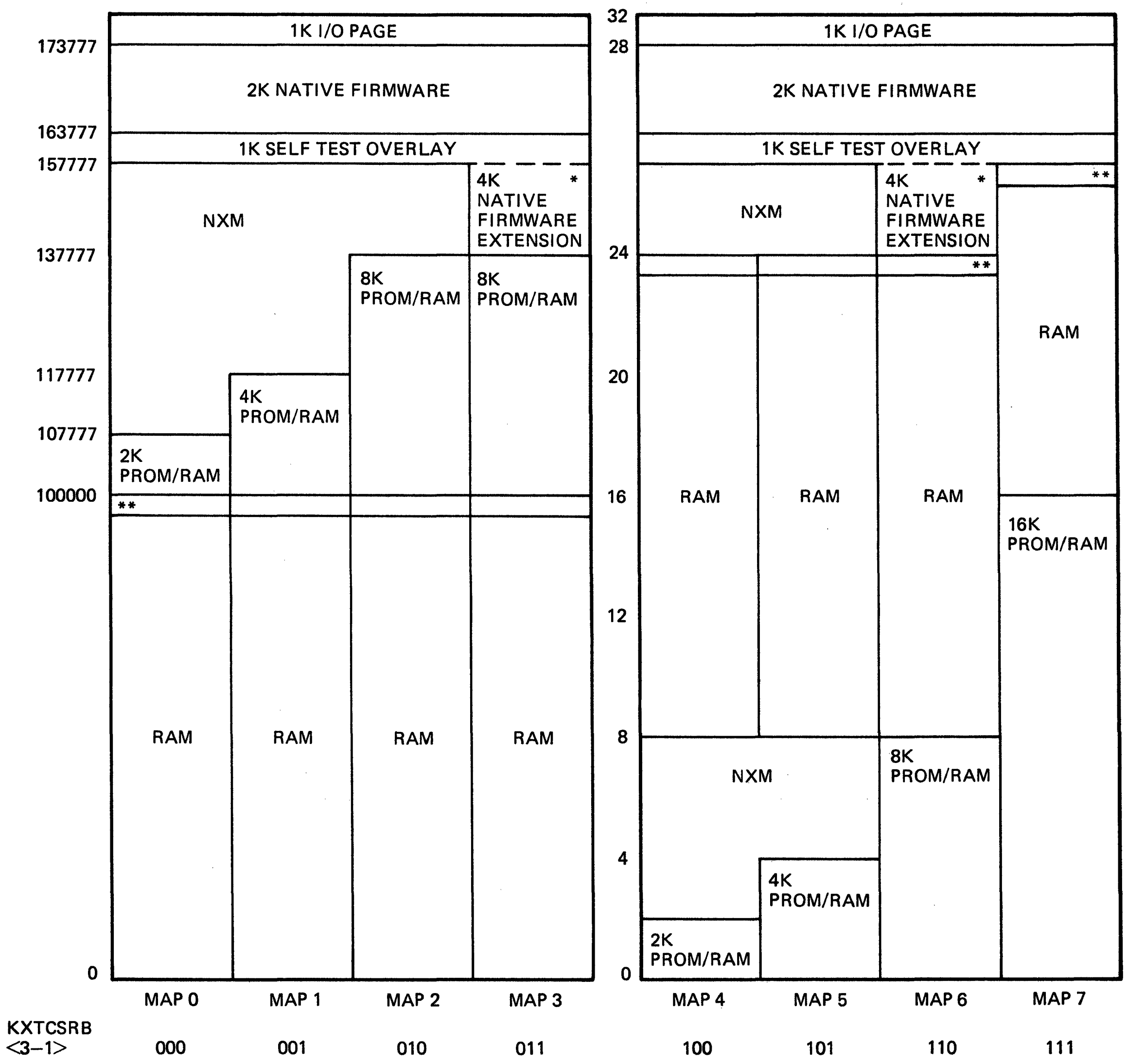


Figure 6-8 Memory Map Jumper Configurations

Table 6-5 Memory Map Jumper Configurations

Memory Map	Jumper Connection M24 to M25	M22 to M23	M20 to M21
0*	In	In	In
1	Out	In	In
2	In	Out	In
3	Out	Out	In
4	In	In	Out
5	Out	In	Out
6	In	Out	Out
7	Out	Out	Out

*Factory shipped configuration.



*NOT CURRENTLY SUPPORTED

**THE TOP 64 BYTES OF NATIVE RAM ARE RESERVED BY NATIVE FIRMWARE

MR-11673

Figure 6-9 Memory Maps

6.3.5 Boot/Selftest Switch

The boot/selftest switch selections are divided into two major functions: first, standalone booting from the user ROM into the application code memory area; second, selftests. Figure 6-10 shows the boot/selftest switch; Table 6-6 describes its functions.

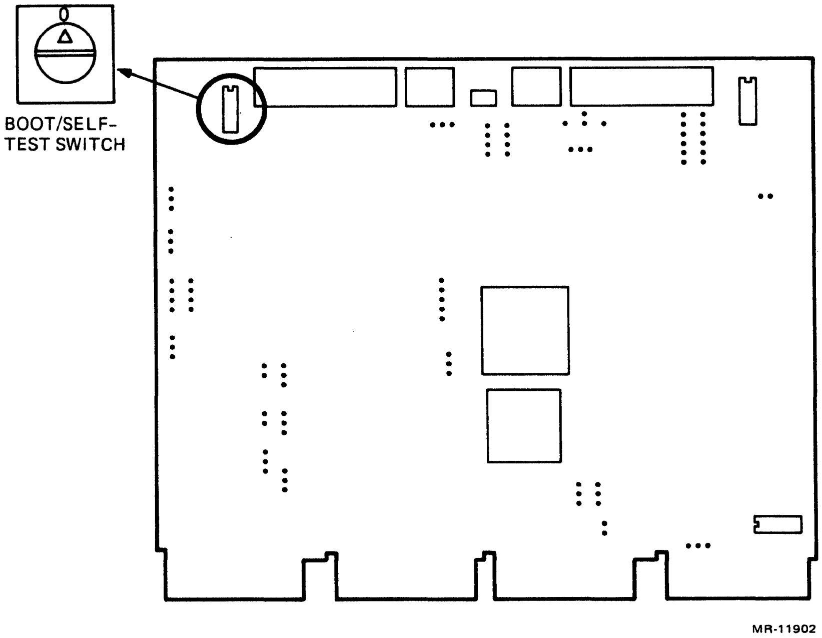


Figure 6-10 Boot/Selftest Switch

Table 6-6 Boot/Selftest Switch Functions

Switch Position	Description
Standalone Boot	
0	User ROM application code is executed; no selftest is performed.
1	Application code in user ROM is executed; a selftest is performed.
2	Application code in user ROM is executed; selftest and ROM test are performed.
3	Boot application çode from TU58 drive unit; a selftest is performed.
4	No auto selftest; enter serial ODT.

Table 6-6 Boot/Selftest Switch Functions (Cont)

Switch Position	Description
System Boot	
5	Selftest is performed; do not boot; wait for boot command from arbiter.
6	No selftest performed; do not boot; wait for boot command from arbiter.
7	Reserved.
Dedicated Test	
8	Used for standalone testing. Selftest is performed; loopback test is run; no application code is run; the RAM is mapped low.
9	Used for standalone testing. Selftest and user ROM test are performed; loopback test is run; no application is code run; the RAM is mapped low.
10*	For use when running XXDP+ diagnostics. Selftest is performed; wait for additional test commands; the RAM is mapped low.
11–15	Reserved.

*Factory shipped configuration.

6.3.6 Q-Bus Base Address (ID Switch)

The jumper between M18 and M19 enables the user to select either a high or low base address range. The system ID switch enables the user to select one of 14 base addresses within a range. The high base address range should be used if a Falcon SBC or more than 8 KXT11-CA SBCs are on the Q-Bus due to a conflict of CPU addressing. Figure 6-11 shows and Table 6-7 lists the Q-Bus base address configurations.

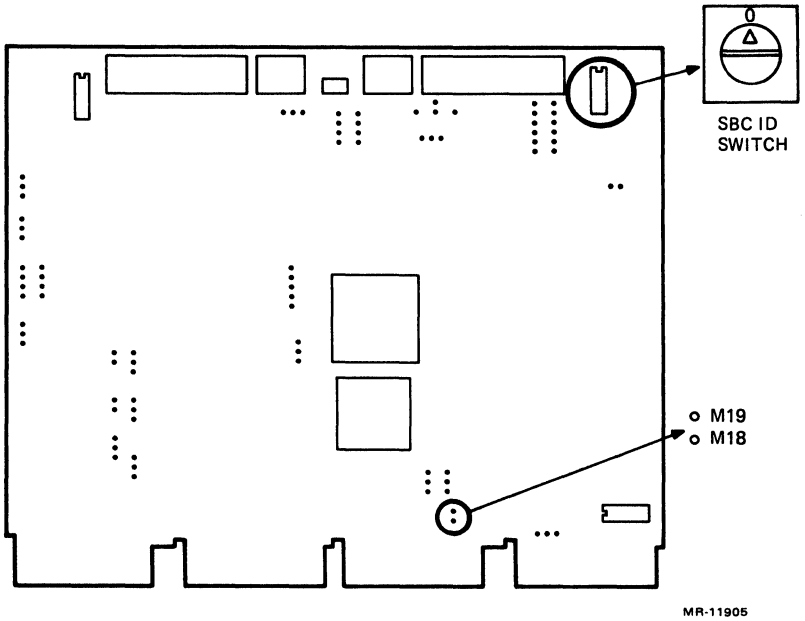


Figure 6-11 Q-Bus Base Address Configurations

Table 6-7 Q-Bus Base Address Configurations

ID Switch Position	Low Range* M18 to M19 Out SBC Base Address	High Range M18 to M19 In SBC Base Address
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‡

* Factory shipped configuration.

† For these two switch positions, the Q-Bus interface is disabled.

‡ Caution – Using these base addresses may conflict with existing Q-Bus devices.

6.3.7 Real-Time Clock Interrupt

The real-time clock interrupt can be set for either 50-Hz or 60-Hz operation. (See Figure 6-12.)

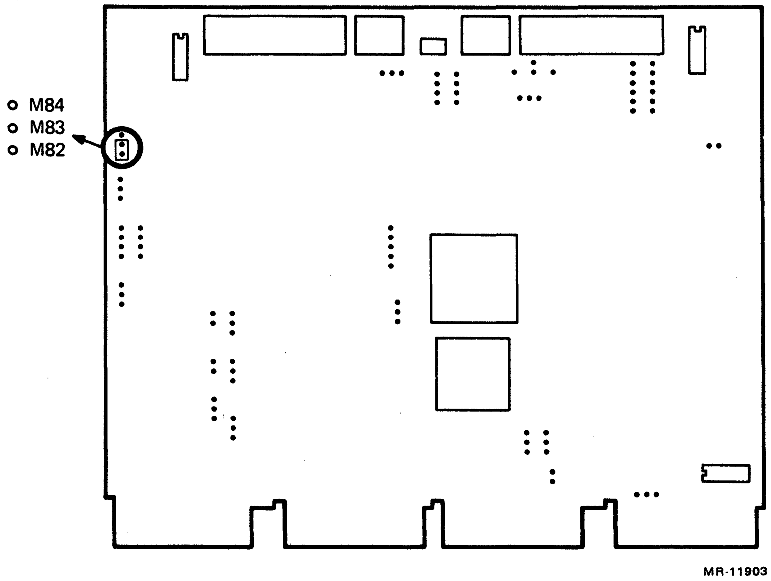


Figure 6-12 Real-Time Clock Interrupt

Jumper Connection	Description
o M84 |o| M83 |o| M82*	Selects the real-time clock 60-Hz interrupt
|o| M84 |o| M83 o M82	Selects the real-time clock 50-Hz interrupt

*Factory shipped configuration.

6.3.8 Break Enable

This interrupt (Figure 6-13) is controlled by the user via the console BREAK key, causing a nonmaskable T-11 trap to restart.

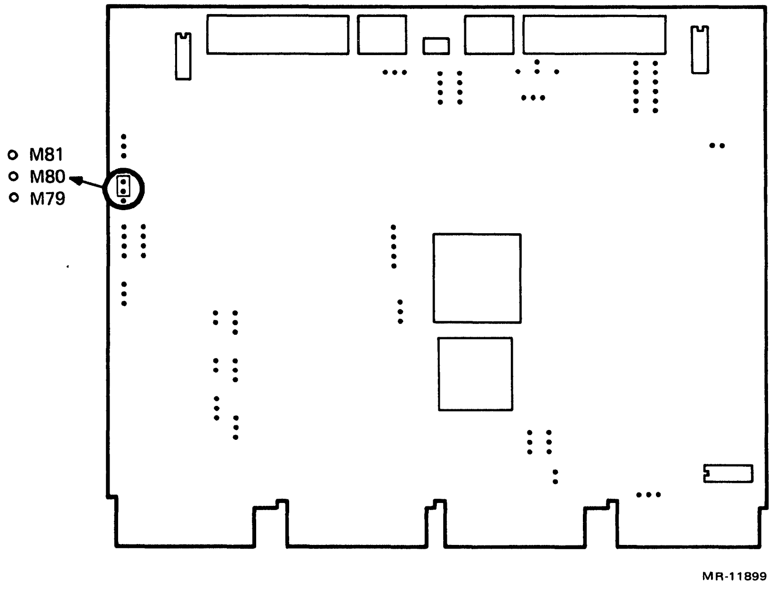


Figure 6-13 Break Enable

Jumper Connection	Description
o M81 o M80* o M79	Selects console SLU1 break key interrupt
o M81 o M80 o M79	Disables SLU1 console BREAK key interrupt

*Factory shipped configuration.

6.3.9 SLU1 Transmitter

SLU1 can be configured to transmit either single-ended (RS423) or differential (RS422) asynchronous serial data out on connector J3. (See Figure 6-14.)

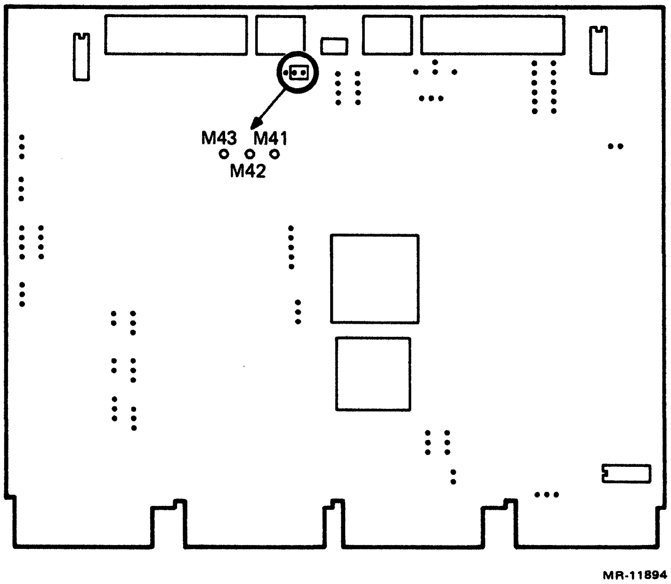


Figure 6-14 SLU1 Transmitter

Jumper Connection	Description
M43 M42 M41* o [o o]	SLU1 console serial output is jumpered for RS423 transmission.
M43 M42 M41 [o o] o	SLU1 console seerial output is jumpered for RS422 transmission.

*Factory shipped configuration.

6.3.10 SLU1 Receiver

SLU1 can be configured to receive either single-ended (RS423) or differential (RS422) asynchronous serial data in on connector J3. (See Figure 6-15.)

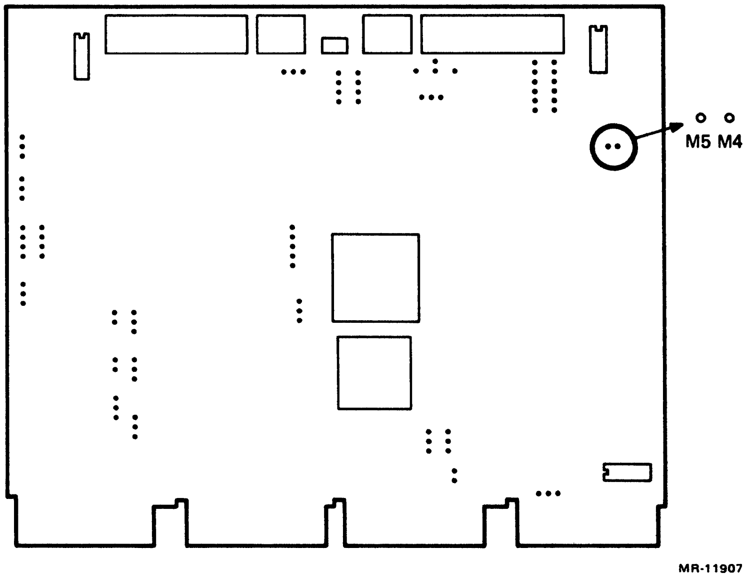


Figure 6-15 SLU1 Receiver

Jumper Connection	Description
M5* M4 o o	SLU1 console serial input single-ended input.
M5 M4 o o	SLU1 console serial input differential input.

*Factory shipped configuration.

6.3.11 SLU1 Baud Rate

SLU1 output baud/bit rate (Figure 6-16, Table 6-8) is hardware programmable from 300 baud to 38400 baud when jumper M78 to M77 is inserted. If this jumper is not inserted, the baud rate is under software control.

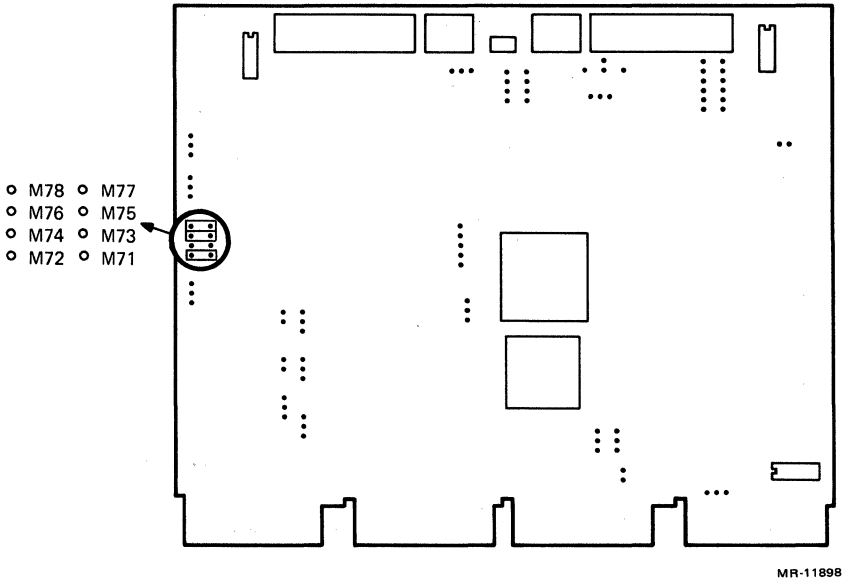


Figure 6-16 SLU1 Baud Rate

Table 6-8 SLU1 Baud Rate

Baud Rate (Bits/S)	Jumper Connection M72 to M71	M74 to M73	M76 to M75
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.

6.3.12 DMA Requests

There are three DMA request lines to the direct transfer controller (DTC). Two request lines come from SLU2 channels A and B. The third request comes from the counter/timer (8036) chip. Only two of the three requests can be jumpered in at one time. (See Figure 6-17.)

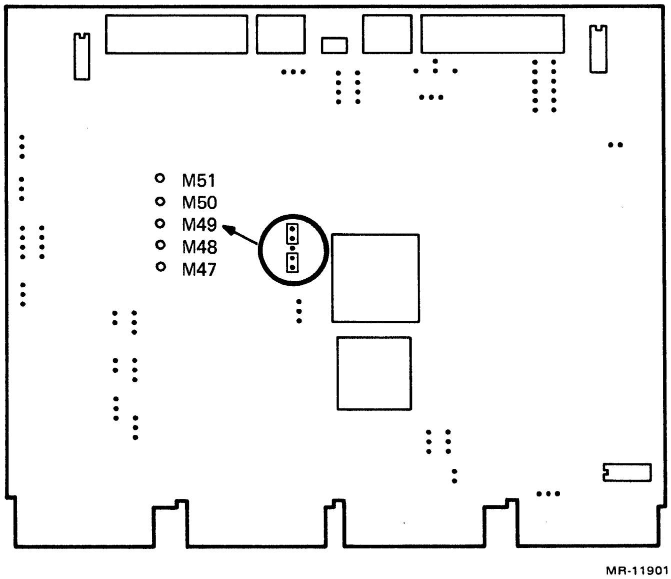


Figure 6-17 DMA Requests

Jumper Connection	Description
o M51* o M50† o M49 o M48 o M47	Jumpers a DMA request from SLU2 channel A.
o M51 o M50† o M49 o M48 o M47	Jumpers a DMA request from the counter/timer chip.
o M51* o M50† o M49 o M48 o M47	Jumpers a DMA request from SLU2 channel B.

* Factory shipped configuration.
† Do not place a jumper from M49 to M50 because this configuration is not supported.

6.3.13 SLU2 Channel A Receiver

SLU2 channel A output (Figure 6-18) can be either single-ended for RS423 receiver or differential for RS422 receiver. Channel A receives a synchronous/asynchronous serial data in on connector J1.

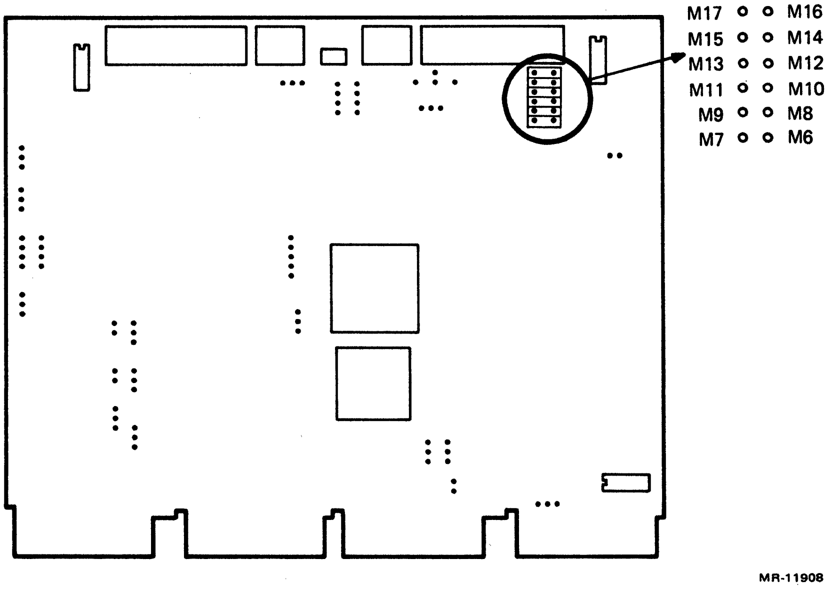


Figure 6-18 SLU2 Channel A Receiver

Jumper Connection†	Description
M17 o o M16 M15 o o M14 M13 o o M12 M11 o o M10 M9 o o M8 M7 o o M6	SLU2 channel A single-ended (RS423) receiver operation
M17 o o M16* M15 o o M14* M13 o o M12* M11 o o M10* M9 o o M8 * M7 o o M6 *	SLU2 differential input (RS422)

* Factory shipped configuration.

† Jumpers are inserted horizontally for a differential signal.

6.3.14 SLU2 Channel B Operation

SLU2 channel B (Figure 6-19) can be configured to receive either driver (differential or single-ended) or party line signals (CCITT R1360) in on connector J2.

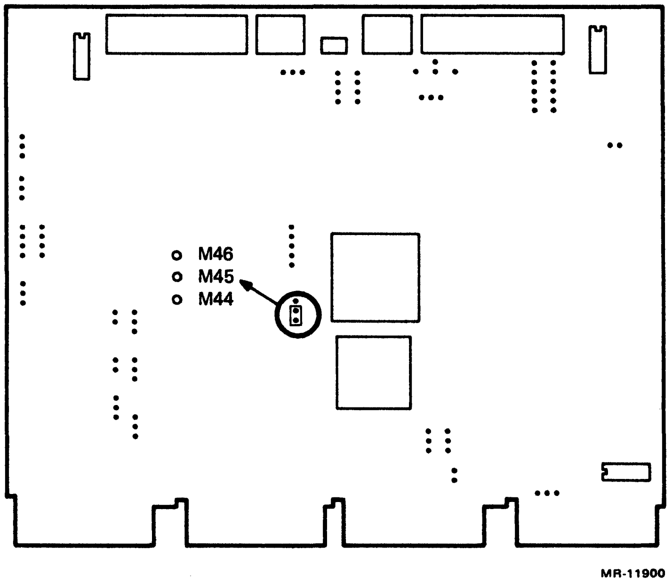


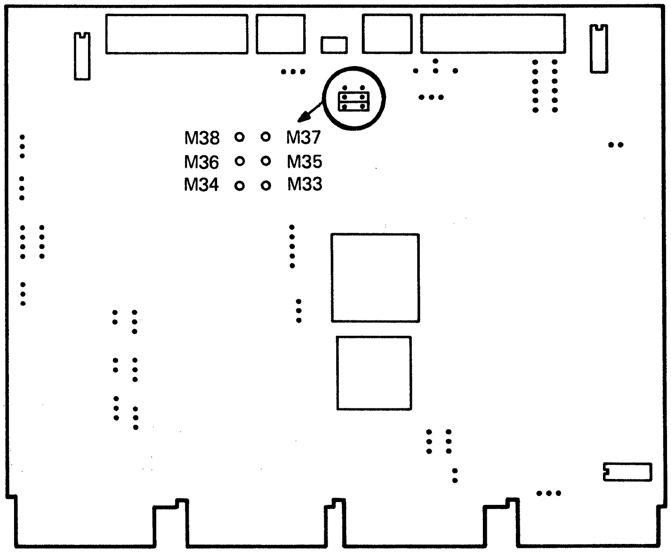
Figure 6-19 SLU2 Channel B Operation

Jumper Connection	Description
M46 o M45 o M44 o	R1360 party line input signal
M46 o M45 o * M44 o	TT104B input signal (RS423)

*Factory shipped configuration.

6.3.15 SLU2 Channel B Receiver

SLU2 channel B input (Figure 6-20) can be configured for RS422 (differential) or RS423 (single-ended) receiver operation. Channel B output can be configured for either RS423 (single-ended), RS422 (differential), or party line (CCITT R1360) operation. When configured for party line operation, a jumper is inserted between M39 and M40 for party line termination.



MR-11895

Figure 6-20 SLU2 Channel B Receiver

Jumper Connection	Description
M38 o o M37 M36 o o M35 M34 o o M33	RS423 receiver operation
M38 o o M37* M36 $\boxed{o \quad o}$ M35 M34 $\boxed{o \quad o}$ M33	RS422 receiver operation
M38 $\boxed{o \quad o}$ M37 M36 o o M35 M34 o o M33	Party line receiver operation

*Factory shipped configuration.

6.3.16 SLU2 Channel B Transmitter

SLU2 channel B transmitter (Figure 6-21) can be selected for one of three operation moves: RS423 (single-ended), RS422 (differential), or party line. The channel transmits synchronous/asynchronous serial data out on connector J2.

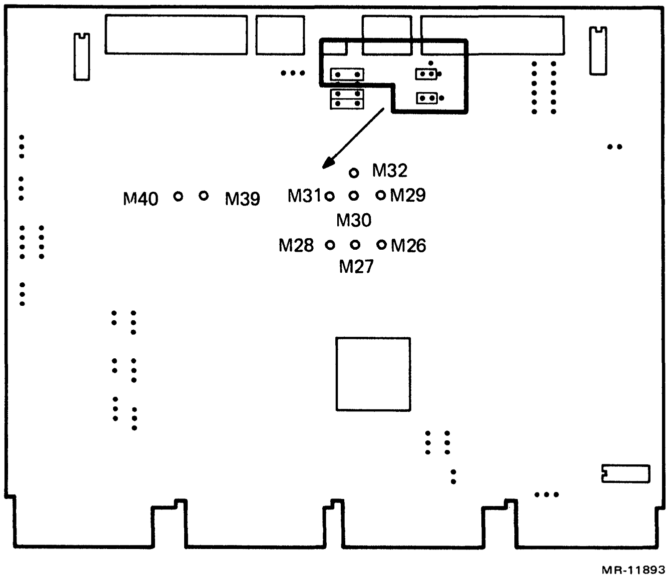
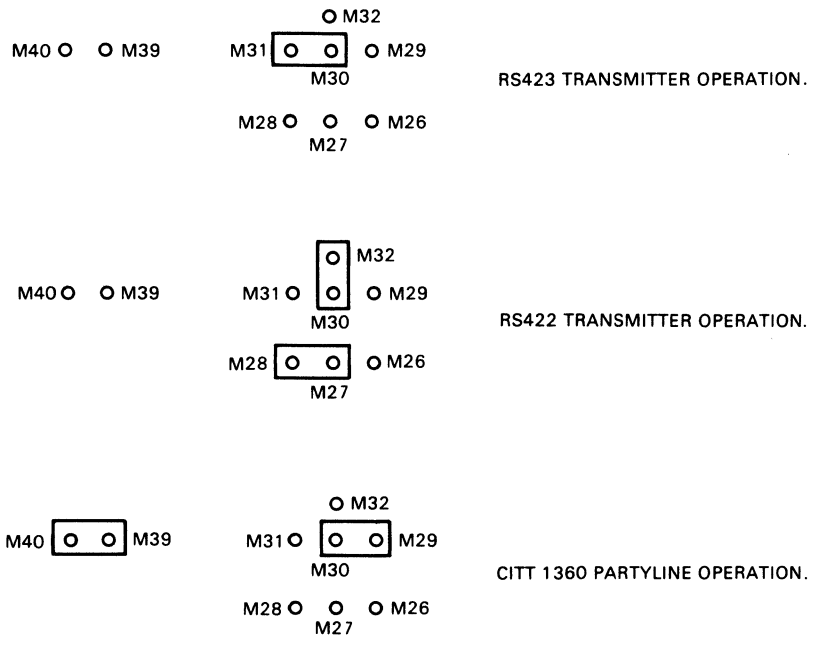


Figure 6-21 SLU2 Channel B Transmitter


M32
M40
M39
M31
M29
M30
M28
M26
M27
RS423 TRANSMITTER OPERATION.
M32
M40
M39
M31
M29
M30
M28
M26
M27
RS422 TRANSMITTER OPERATION.
M32
M40
M39
M31
M29
M30
M28
M26
M27
CITT 1360 PARTYLINE OPERATION.

NOTE: IF TERMINATION IS NEEDED M39 TO M40 IS JUMPERED.

MR-12619

6.4 POWER SUPPLY

The choice of power supply is controlled by the size of the system and packaging requirements. An important consideration is the performance of the supply during power-up and power-down. All Digital Equipment Corporation power supplies listed in the *Microcomputer Interfaces Handbook* are compatible with the Q-Bus protocol which allows dependable operation with no loss of data when using battery backed-up memories.

6.5 BACKPLANE INSTALLATION

The KXT11-CA board can be inserted into either of the two currently available types of standard backplane assemblies made by Digital Equipment Corporation. One backplane assembly is the Q/Q22-CD type (H9273), and the other backplane assembly is the Q/Q22-Q/Q22 type (H9275). An important point to remember when installing a KXT11-CA SBC into either of these backplane assemblies is the bus grant continuity. Both types of backplanes have unique bus grant daisy-chain arrangements and are priority structured. The highest priority propagates away from the top-most slot downward. The user can ensure proper grant continuity for all backplane-installed boards by inserting each device in the first slot physically below a previously installed board referenced from the assembly top.

The Q/Q22-CD backplane assembly is shown in Figure 6-22. It can accept both double-height and quad-height boards. If double-height boards are inserted, the grant continuity is supported only for slots A and B. Any double-height boards inserted in slots C and D are not supported. The KXT11-CA board grant continuity is supported with this backplane assembly.

The Q/Q22-Q/Q22 backplane assembly is shown in Figure 6-22 also. It can accept both double-height and quad-height boards. Double-height board grant continuity is supported for both slots A and B and C and D. In addition, quad-height KXT11-CA board grant signal continuity is supported on the KXT11-CA board with mounted jumpers W1 and W2. If, however, a quad-height KXT11-CA board is to be installed at a lower priority than an existing double-height board, then a grant card (G7272) must be installed. The grant card and backplane are shown in Figure 6-23.

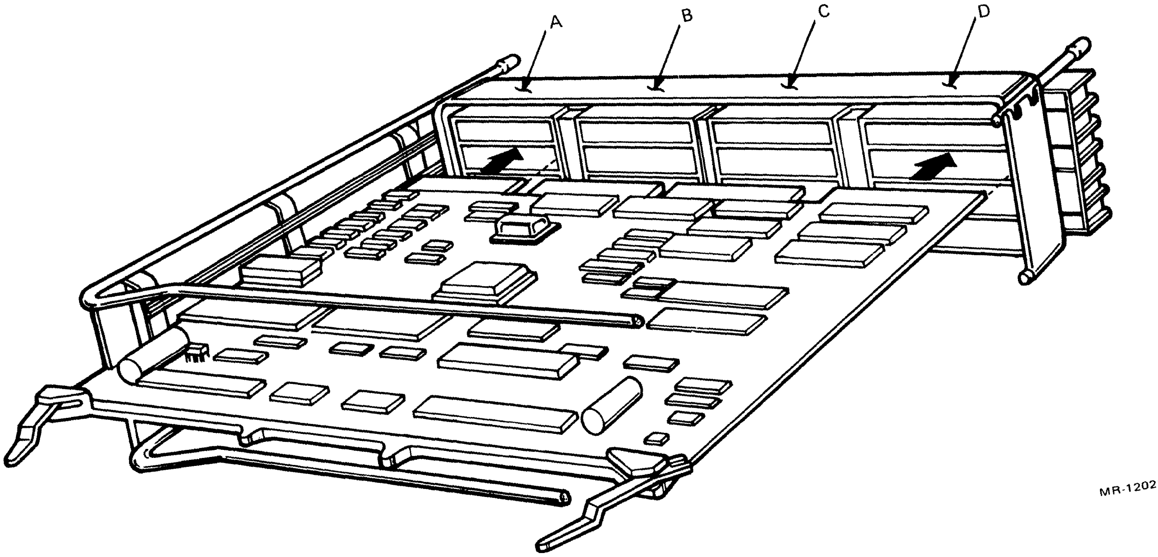


Figure 6-22 Q/Q22-CD Backplane Assembly

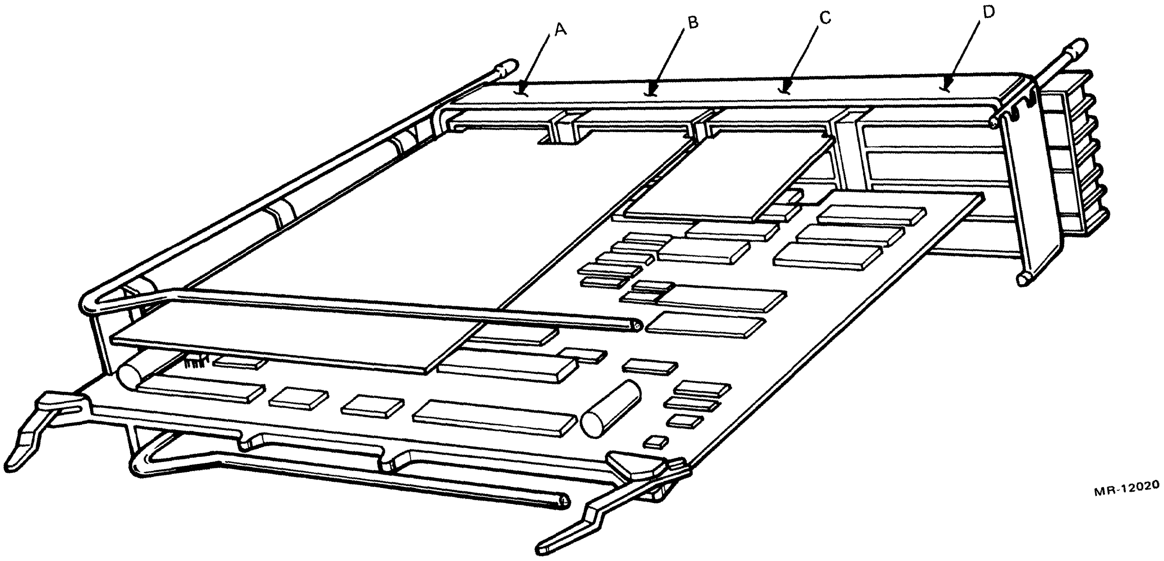


Figure 6-23 Q/Q22–Q/Q22 Backplane Assembly With Grant Card

6.6 BACKPLANE PIN IDENTIFICATION

Table 6-9 lists backplane pin connections for the KXT11-CA SBC, pin identification and signal names unique to the KXT11-CA SBC, and standard Q-Bus backplane names assigned to each pin. Although the signal names may differ, the SBC is completely Q-Bus-compatible, with the exception of bus refresh transaction (BREF), which is not performed by the KXT11-CA SBC.

Table 6-9 KXT11-CA Backplane Pin Identification

Backplane Pin	KXT11-CA Signal Function	Q-Bus Signal Name
Side 1 (Component Side)		
AA1	Not connected	BIRQ5 L
AB1	Not connected	BIRQ6 L
AC1	Bus terminator	BDAL16 L
AD1	Bus terminator	BDAL17 L
AE1	Not connected	SSPARE1
AF1	Not connected	SSPARE2
AH1	Not connected	SSPARE3
AJ1	GND	GND
AK1	Not connected	MSPAREA
AL1	Not connected	MSPAREA
AM1	GND	GND
AN1	BDMR L	BDMR L
AP1	BHALT L	BHALT L
AR1	Not connected	BREF L
AS1	Not connected	+12B
AT1	GND	GND
AU1	Not connected	PSPARE1
AV1	+5 VB (battery)	+5B
BA1	BDCOK H	BDCOK H
BB1	BPOK H	BPOK H
BC1	BDAL18 L	SSPARE4
BD1	BDAL19 L	SSPARE5
BE1	BDAL20 L	SSPARE6
BF1	BDAL21 L	SSPARE7
BH1	START L	SSPARE8
BJ1	GND	
BK1	Not connected	MSPAREB
BL1	Not connected	MSPAREB
BM1	GND	
BN1	BSACK L	BSACK L
BP1	Not connected	BIRQ7 L
BR1	Not connected	BEVNT L
BS1	Not connected	+12B
BT1	GND	GND
BU1	Not connected	PSPARE2
BV1	+5 V	+5 V

Table 6-9 KXT11-CA Backplane Pin Identification (Cont)

Backplane Pin	KXT11-CA Signal Function	Q-Bus Signal Name
Side 2 (Solder Side)		
AA2	+5 V	+5 V
AB2	Not connected	−12 V
AC2	GND	GND
AD2	+12 V	+12 V
AE2	BDOUT L	BDOUT L
AF2	BRPLY L	BRPLY L
AH2	BDIN L	BDIN L
AJ2	BSYNC L	BSYNC L
AK2	BWTBT L	BWTBT L
AL2	BIRQ4 L	BIRQ4 L
AM2	BIAKI L	BIAKI L
AN2	BIAKO L	BIAKO L
AP2	BBS7 L	BBS7 L
AR2	BDMGI L	BDMGI L
AS2	BDMGO L	BDMGO L
AT2	BINIT L	BINIT L
AU2	BDAL0 L	BDAL0 L
AV2	BDAL1 L	BDAL1 L
BA2	+5 V	+5 V
BB2	Not connected	−12 V
BC2	GND	GND
BD2	+12 V	+12 V
BE2	BDAL2 L	BDAL2 L
BF2	BDAL3 L	BDAL3 L
BH2	BDAL4 L	BDAL4 L
BJ2	BDAL5 L	BDAL5 L
BK2	BDAL6 L	BDAL6 L
BL2	BDAL7 L	BDAL7 L
BM2	BDAL8 L	BDAL8 L
BN2	BDAL9 L	BDAL9 L
BP2	BDAL10 L	BDAL10 L
BR2	BDAL11 L	BDAL11 L
BS2	BDAL12 L	BDAL12 L
BT2	BDAL13 L	BDAL13 L
BU2	BDAL14 L	BDAL14 L
BV2	BDAL15 L	BDAL15 L

6.7 EXTERNAL CABLES

The SBC has a 40-pin connector (J4) for an external interface with the programmable I/O interface, a 40-pin connector for full modem support interface (J1), and two 10-pin connectors (J2 and J3) for the external interface of the serial line units (SLUs). The locations of these connectors on the SBC are shown in Figure 6-22. The requirements to interface with these connectors are defined in the following paragraphs.

6.7.1 Parallel I/O Interface (J4)

The I/O signals are buffered and are capable of driving up to 50 feet (maximum) of flat ribbon or round cable with a 40-pin AMP contact housing at each end. The following two cables are compatible with the module connector.

1. BC05 L mirror image cable
2. BC06 R shielded ribbon cable

Figure 6-24 shows the PIO connector pin assignments.

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

NOTE: SG = SIGNAL GROUND
A0 - A7 = PORT A
B0 - B7 = PORT B
C0 - C7 = PORT C

MR-12615

Figure 6-24 PIO Connector Pin Assignments

6.7.2 Serial I/O Interfaces (J1, J2 and J3)

Each serial line unit (SLU) is compatible with EIA RS232-C and EIA RS423 serial type interfaces. SLU1 interfaces through J3, SLU2 channel A interfaces through J1, while SLU2 channel B interfaces through J2. When a 20 mA current loop device is needed, the DLV11-KA option must be used.

The user provides the interconnecting cables. The following list describes some standard Digital cables for use with the KXT11-CA SBC and also provides some information to help the user design cables.

BC20N-05 5-foot EIA RS232-C null modem cable to directly interface with the EIA terminal (2 × 5 pin AMP female to RS232-C female)

BC21B-05 5-foot EIA RS232-C modem cable to interface with modems and acoustic couplers (2 × 5 pin AMP female to RS232-C male)

BC20M-50 50-foot EIA RS422 or RS423 cable for high throughput transmission (19.2 K baud) between two KXT11-CA computers (2 × 5 pin AMP female to 2 × 5 pin AMP female)

When designing a cable for the KXT11-CA SBC, the user should consider the following points.

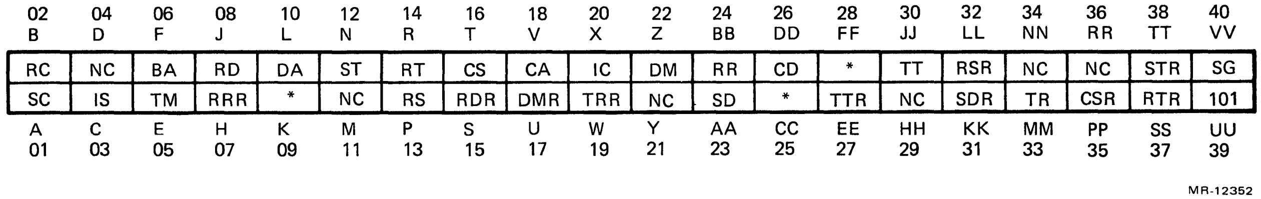
1. The receivers on the KXT11-CA SBC have differential inputs. Therefore, when designing an RS232-C or RS423 cable, RECEIVE DATA– (pin 7 on the 2 × 5 pin AMP connector) must be tied to signal ground (pins 2, 5, or 9) in order to maintain correct EIA levels. RS422 uses both RECEIVE DATA+ and RECEIVE DATA–.

2. To directly connect to a local EIA RS232-C terminal, use a null modem. To design the null modem into the cable, a user must switch RECEIVE DATA (pin 2) with TRANSMITTED DATA (pin 3) on the RS232-C male connector as shown in Appendix H.

3. To mate to the 2 × 5 pin connector block, the following parts are needed.

Cable receptacle	AMP PN 87133-5 DEC PN 12-14268-02
Locking clip contacts	AMP PN 87124-1 DEC PN 12-14267-00
Key pin (pin 6)	AMP PN 87179-1 DEC PN 12-15418-00

The pin assignments for connector J2 and J3 are shown in Appendix F, Figure F-2. The pin assignments for connector J1 are shown in Figure 6-25. Table 6-10 lists compatible I/O signal names for J1.



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

MR-12352

Figure 6-25 SLU2 Connector (J1) Pin Assignments

Notes to Figure 6-25:

1. Pins denoted by an asterisk (*) are driven by dummy generators and hold the RL (TT140), LL (TT141), and SS (TT116) signals in the off state.

2. NC indicates no connection.

3. R attached to a two-letter code implies the RETURN side of a differential driver or receiver.

4. Signals marked with a $ are not supported.

5. Signals marked with a # can be defined by the user at the expense of another signal, i.e., circuit IS can be redefined to mean SR and circuit TM can be redefined to mean SI, or only circuit IS could be redefined to mean SF.

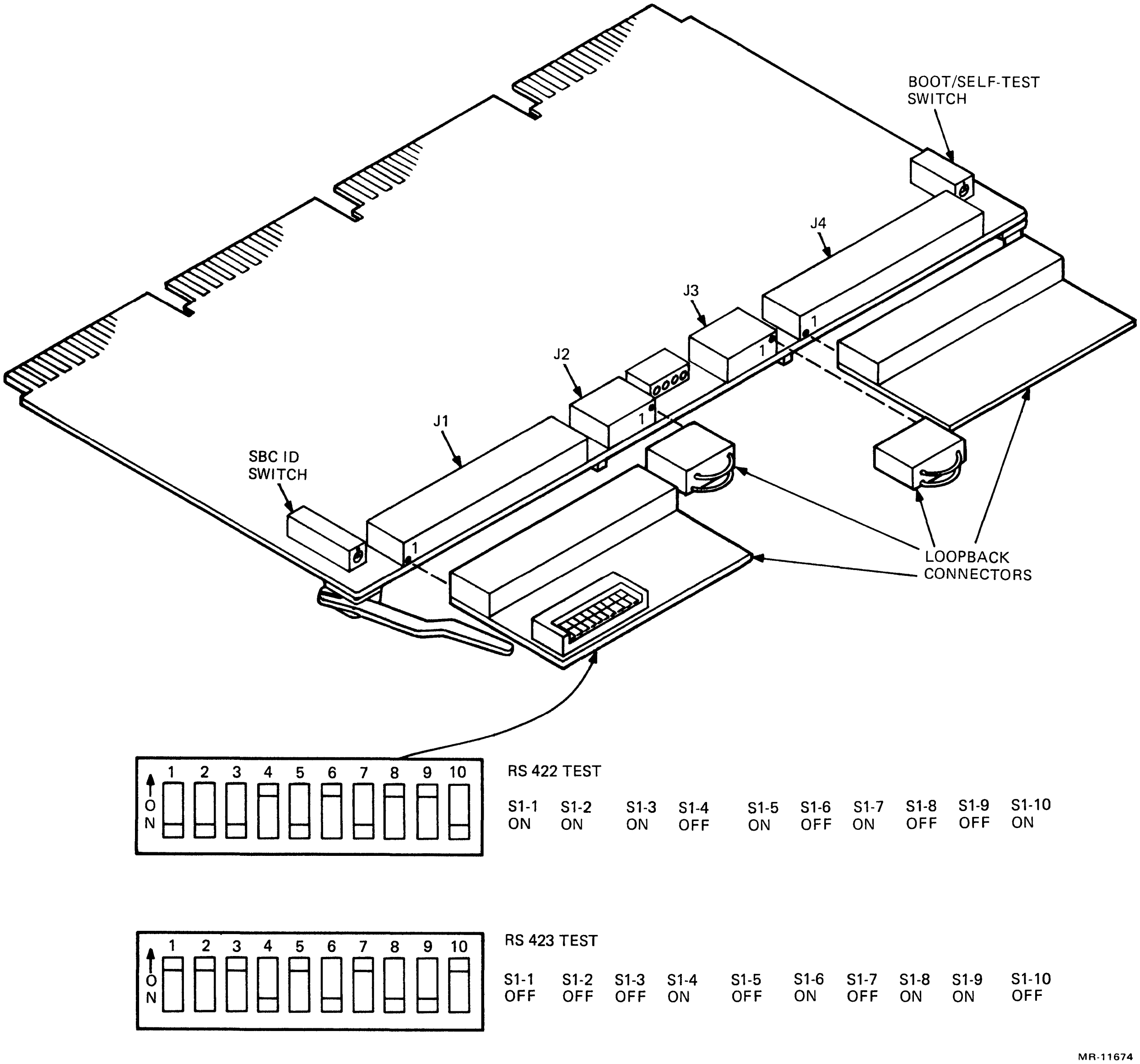
Table 6-10 Compatible I/O Signal Names

J1 (SLU2, Channel A)	RS449 Signal	CCITT Logical Equivalent	RS232-C Signal
40, VV	SB Signal ground	102	AB Signal ground
1, A	SC Send common	102a	-
2, B	RC Receive common	102b	-
3, C	#IS Terminal in service	-	-
20, X	IC Incoming call	125	CE Ring indicator
33, M	TR Terminal ready	108/2	CD Data terminal ready
22, Z	DM Data mode	107	CC Data set ready
23, AA	SD Send data	103	BA Transmitted data
8, J	RD Receive data	104	BB Received data
30, JJ	TT Terminal timing	113	DA DTE signal element timing (transmitter)
12, N	ST Send timing	114	DB DCE signal element timing (transmitter)
14, R	RT Receive timing	115	DD DCE receiver signal element timing
13, P	RS Request to send	105	CA Request to send
16, T	CS Clear to send	106	CB Clear to send
24, BB	RR Receiver ready	109	CF Received line signal detector
5, E	#TM Test mode	142	-
N/A	-	101 Signal ground	-
N/A	$SS Select standby	116	-
N/A	SB Standby indicator	117	-
	LL Local loopback	141	-
N/A	$RL Remote loopback	140	-
N/A	$SQ Signal quality	110	CB Signal quality
N/A	$NS New signal	-	-
See Note	#SF Select frequency	126 Select transmit frequency	-
See Note	#SR Signaling rate selector	111 DTE data signal rate selector	CH DTE data signal rate selector
See Note	#SI Signaling rate indicator	112 DCE data signal rate selector	CI DCE data signal rate selector

Note:
A dollar sign ($) preceding a signal name indicates the signal is unsupported. A pound sign (#) preceding a signal name indicates the signal can be defined by the user at the expense of another signal. For example, signal IS can be redefined to mean SR and signal TM can be redefined to mean SI, or only signal IS could be redefined to mean signal SF.

6.8 LOOPBACK CONNECTORS

Three loopback connectors – part numbers H3021, H3022 and H3270 – are used in testing the KXT11-CA single-board computer. Two 10-pin loopback connectors (H3270) are used: (1) SLU2 channel B sync/async I/O connector J2, and (2) SLU1 console I/O connector J3. The 40-pin loopback connector (H3022) is used with SLU2 channel A sync/async I/O connector J1. Note that the J1 connector can be configured, with 10 switches, for RS422 or RS423 loopback operation. The 40-pin loopback connector (H3021) is used with the parallel I/O connector J4. Insertion of the loopback connectors is shown in Figure 6-26. Users can design and build their own loopback connectors that are compatible. (See Appendix H for information.)



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 6-26 Loopback Connectors

6.9 KXT11-CA SBC DIAGNOSTIC TESTING PROCEDURES

After properly booting the XXDP+ diagnostic software, the user is required to answer all the console prompts. Answering the console prompts builds entries into a table describing each KXT11-CA SBC under test. The console prompts are as follows:

1. Enter the date and answer the LSI and 50-Hz questions.
2. Type "R CZKTCA" to load the KXT11-CA test software.
3. After the test software is loaded into the arbiter, prompt "DS>" appears.
4. Type "START", then <CR>.
5. Answer the question, "CHANGE HW(L)?" with "Y", then <CR>.
6. Answer the question, "# UNITS (D) ?" with the number of units in the system, then <CR>.
7. The prompt "UNIT 0" appears.
8. Answer the question, "SBC ID SWITCH SETTING (D) ?" with the setting of the SBC under test, then <CR>.
9. Answer the question, "IS THE IOP'S LSI-11 ADDRESS IN THE LOW ADDRESS RANGE (Y/N) ?" with "Y" if the jumper is in or "N" if it is out, then <CR>.
10. Answer the question, "LOOP-BACK CONNECTOR ON SERIAL PORT #1 (L)" with "Y" or "N", then <CR>.
11. Answer the question, "CHANNEL A LOOP-BACK CONNECTOR ON SERIAL PORT #2 (L)" with "Y" or "N", then <CR>.
12. Answer the question, "CHANNEL B LOOP-BACK CONNECTOR ON SERIAL PORT #2 (L) " with "Y" or "N", then <CR>.
13. Answer the question, "LOOP-BACK CONNECTOR ON PARALLEL PORT (L)" with "Y" or "N", then <CR>.
14. Answer the question, "IS SLU2 SERIAL PORT CONFIGURED FOR DMA OPERATION (L)" with "Y" or "N", then <CR>.
15. Answer the question, "TEST USER ROM (L)" with "Y" or "N", then <CR>.

After the last <CR>, in step 15, the test software is immediately invoked.

6.10 ERROR INFORMATION

The results of the KXT11-CA selftest and firmware state will be displayed by the edge mounted LEDs. The LEDs are used to indicate how far the SBC got into the selftest process. Refer to Figure 6-27 for the LED display sequence and error codes.

On power-up or reinitialization the LEDs are lit for approximately one-half second to demonstrate they work correctly. The LEDs flicker as they cycle through the selftests. If L1 is off after the completion of the selftests, an error was encountered and is being displayed. The boot sequence continues if a nonfatal error is detected, but the LEDs indicate the state in which the first error was detected. If the LEDs encounter a fatal error, the boot sequence stops. If the SBC remains in the primary bootstrap or wait state, this indicates a possible error in booting or master/IOP communications, respectively.

LED State*				**Definition**
L1	**L2**	**L3**	**L4**	
x	x	x	x	All LEDs on for about 0.5 s only during power-up or reinitialization
x	x	x	x	Fatal SBC selftest error (SBC not functional)
x	o	o	o	Nonfatal SBC selftest error
x	o	o	x	User ROM low byte test failed or user RAM test failed
x	o	x	o	User ROM high byte test failed
x	o	x	x	Serial loopback of SLU1 failed
x	x	o	o	Serial loopback of SLU2 channel A failed
x	x	o	x	Serial loopback of SLU2 channel B failed
x	x	x	o	Parallel loopback failed
o	x	x	x	Auto selftests running
o	x	x	o	Loopback tests running
o	x	o	x	Q-Bus ODT mode
o	x	o	o	Serial ODT mode
o	o	x	x	Waiting for command
o	o	x	o	Performing DMA load
o	o	o	x	TU58 primary bootstrap executing
o	o	o	o	Executing non-native code

Quick LED Reference				
L1	**L2**	**L3**	**L4**	
o	+	+	+	No errors, state displayed
o	o	o	o	Normal, application running state
x	+	+	+	Error detected
x	x	x	x	Fatal error detected

x = On, o = Off, + = Either

* L1 lit indicates error display; L1 unlit indicates state display. L2 through L4 lit indicate state of SBC.

Figure 6-27 LED Error Display

The not-running native code is the normal state of the SBC. This state is displayed when the application code or a secondary loader is running and no errors were previously displayed. State changes are not displayed by the LEDs if a previous error is being displayed.

There are three levels of error messages that may be issued by a diagnostic to the system arbiter: general, basic, and extended.

General error messages are always printed unless the IER flag is set (see the *XXDP+ User's Manual*). The general error message format follows:

```
NAME   TYPE   NUMBER   ON UNIT NUMBER   TEST NUMBER PC:XXXXXX
ERROR MESSAGE
WHERE; NAME = DIDAGNOSTIC NAME
                  TYPE = ERROR TYPE (SYS FATAL, HARDWARE OR SOFTWARE)
                                                NUMBER = ERROR NUMBER
                                                  UNIT NUMBER = 0 – N
                TEST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
                        PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL
```

Basic error messages are messages that contain some additional information about the error. These are always printed unless the IER or IBR flags are set (see the *XXDP+ User's Manual* for details). These messages are printed after the associated general messages.

Extended error messages contain supplementary error information such as register contents or good/bad data. These are always printed unless the IER, IBR, or IXR flags are set (see the *XXDP+ User's Manual* for details). Messages are printed after the associated general error message and any associated basic error.

APPENDIX A
ADDRESSING MODES AND INSTRUCTION SET

A.1 INTRODUCTION

This appendix provides a detailed description of addressing modes and of individual instructions. The description of addressing modes is divided into six major topics.

1. Single-operand addressing – One part of the instruction word specifies the registers; the remaining part provides information for locating the operand.

2. Double-operand addressing – Part of the instruction word specifies the registers; the remaining parts provide information for locating two operands.

3. Direct addressing – The operand is the content of the selected register.

4. Deferred (indirect) addressing – The content of the selected register is the address of the operand.

5. Use of the program counter (PC) as a general-purpose register – The PC is unique from other general-purpose registers. Whenever the processor retrieves an instruction, it automatically advances the PC by two. By combining this automatic advancement of the PC with four of the basic addressing modes, four special PC modes are produced – immediate, absolute, relative, and relative deferred.

6. Use of the stack pointer (SP) as a general-purpose register – The SP can be used for stack operations.

NOTE

Instruction mnemonics and address mode symbols are sufficient for writing assembly language programs. The programmer need not be concerned about conversion to binary digits; this is accomplished automatically by the assembler program.

A.2 ADDRESSING MODES

Data stored in memory must be accessed and manipulated. Data handling is specified by a KXT11-CA instruction (MOV, ADD, etc.) that usually specifies the following.

1. The function to be performed (operation code).

2. A general-purpose register to be used when locating the source and/or destination operand.

3. An addressing mode that specifies how the selected register(s) is/are to be used.

Most data handled by a computer is structured (in character strings, arrays, lists, etc.). KXT11-CA addressing modes allow efficient and flexible handling of structured data.

The general-purpose registers may be used with an instruction in any of the following four ways.

1. As accumulators – The data to be manipulated resides within the register.

2. As pointers – The content of the register is the operand address, rather than the operand itself.

3. As pointers that automatically step through memory locations – Automatically stepping forward through consecutive locations is known as autoincrement addressing; automatically stepping backward is known as autodecrement addressing. These modes are particularly useful for processing tabular or array data.

4. As index registers – The register contents and the word following the instruction are summed to produce the address of the operand. This allows easy access to variable entries in a list.

The register arrangement is an important microprocessor feature that should be considered in conjunction with the addressing modes. There are six general-purpose registers (R0–R5), a hardware stack pointer (SP) register (R6), and a program counter (PC) register (R7).

Registers R0–R5 are not dedicated to any specific function; their use is determined by the instruction that is decoded.

1. They can be used for operand storage. For example, the contents of two registers can be added and stored in another register.

2. They can contain the address of an operand or serve as pointers to the address of an operand.

3. They can be used for the autoincrement or autodecrement features.

4. They can be used as index registers for convenient data and program access.

The KXT11-CA SBC also has instruction-addressing mode combinations that facilitate temporary data storage structures. These combinations can be used for conveniently handling data that must be accessed frequently. This is known as stack manipulation. The register that keeps track of stack manipulation is the stack pointer (SP). Any register can be used as a stack pointer under program control; however, certain instructions associated with subroutine linkage and interrupt service automatically use register R6 as a hardware stack pointer, and therefore, R6 is frequently referred to as the SP.

- The stack pointer keeps track of the latest entry on the stack.

- The stack pointer moves down as items are added to the stack and moves up as items are removed. It always points to the top of the stack.

- The hardware stack is used during trap or interrupt handling to store information and allow the processor to return to the main program.

Register R7 is used by the processor as its program counter (PC) and should not be used as a stack pointer or accumulator. Whenever an instruction is fetched from memory, the program counter is automatically incremented by two to point to the next instruction word.

A.2.1 Single-Operand Addressing

The instruction format for all single-operand instructions (such as clear, increment, and test) is illustrated in Figure A-1.

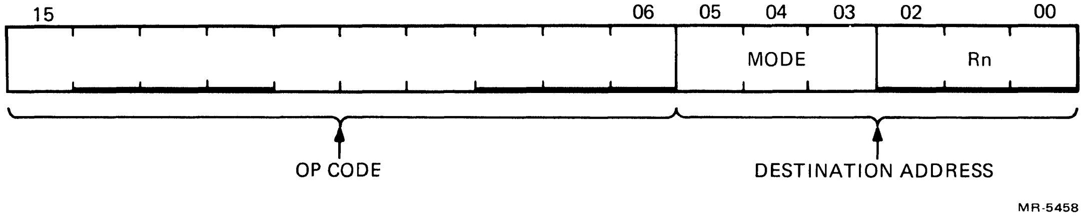


Figure A-1 Single Operand Addressing

Bits 15–6 specify the operation code that defines the type of instruction to be executed. Bits 5–0 form a six-bit field called the destination address field that consists of two subfields.

1. Bits 0–2 specify which of the eight general-purpose registers is to be referenced by the instruction word.

2. Bits 3–5 specify how the selected register will be used (address mode). Bit 3 is set to indicate deferred (indirect) addressing.

A.2.2 Double-Operand Addressing

Operations that imply two operands (such as add, subtract, move, and compare) are handled by instructions that specify two addresses. The first operand is called the source operand; the second operand is called the destination operand. Bit assignments in the source and destination address fields may specify different modes and different registers. The instruction format for the double-operand instruction is illustrated in Figure A-2.

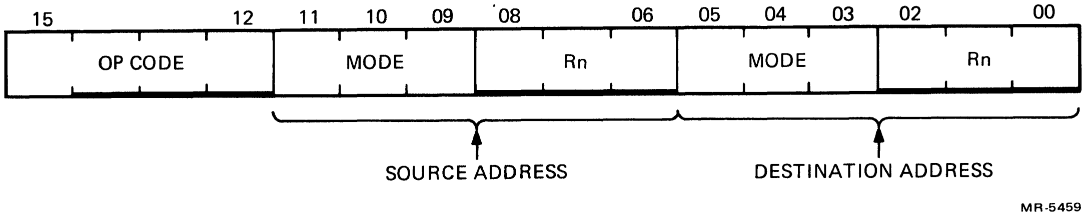


Figure A-2 Double Operand Addressing

The source address field is used to select the source operand, the first operand. The destination is used similarly and locates the second operand and the result. For example, the instruction ADD A, B adds the contents (source operand) of location A to the contents (destination operand) of location B. After execution, B will contain the result of the addition; the contents of A will be unchanged.

Examples in this chapter use the sample KXT11-CA instructions listed in Table A-1. See Paragraph A.3 for a complete list of the KXT11-CA instructions.

Table A-1 Sample KXT11-CA Instructions

Mnemonic	Description	Octal Code
CLR	Clear (zero the specified destination)	0050DD
CLRB	Clear byte (zero the byte in the specified destination)	1050DD
INC	Increment (add one to the contents of the destination)	0052DD
INCB	Increment byte (add one to the contents of the destination byte)	1052DD
COM	Complement (replace the contents of the destination by its logical complement; each zero bit is set and each one bit is cleared)	0051DD
COMB	Complement byte (replace the contents of the destination byte by its logical complement; each zero bit is set and each one bit is cleared)	1051DD
ADD	Add (add source operand to destination operand and store the result at destination address)	06SSDD

DD	= destination field (6 bits)
SS	= source field (6 bits)
()	= contents of

A.2.3 Direct Addressing

Table A-2 summarizes the four basic modes used with direct addressing. Figures A-3, A-4, A-5, and A-6, which follow the table, illustrate these four modes.

Table A-2 Direct Addressing Modes

Mode	Name	Assembler Syntax	Function
0	Register	Rn	Register contains operand.
2	Autoincrement	(Rn)+	Register is used as a pointer to sequential data, then incremented.
4	Autodecrement	–(Rn)	Register is decremented and then used as a pointer.
6	Index	X(Rn)	Value X is added to (Rn) to produce address of operand. Neither X nor (Rn) is modified.

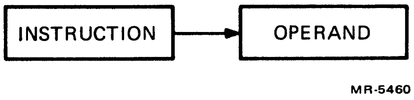


Figure A-3 Mode 0 Register

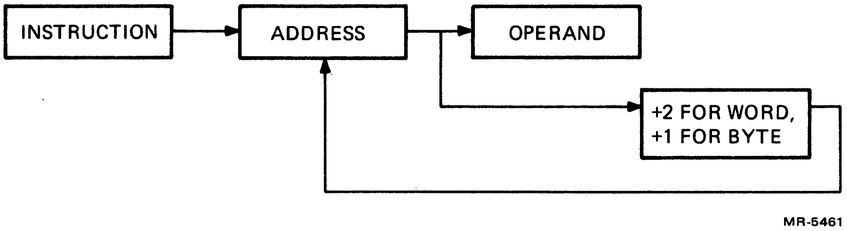


Figure A-4 Mode 2 Autoincrement

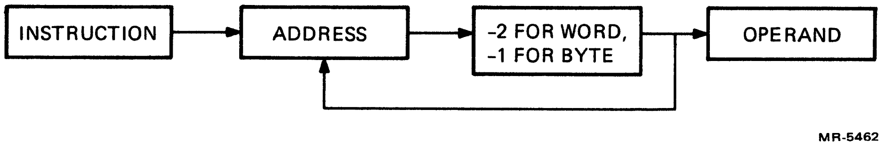


Figure A-5 Mode 4 Autodecrement

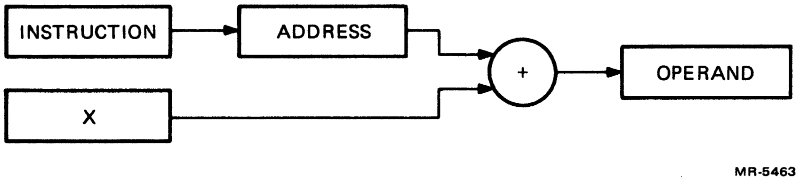


Figure A-6 Mode 6 Index

A.2.3.1 Register Mode (Mode 0) – With register mode, any of the general-purpose registers may be used as simple accumulators, and the operand is contained in the selected register. Because they are hardware registers, within the processor, the general-purpose registers operate at high speeds and provide speed advantages when used for operating on frequently accessed variables. The assembler interprets and assembles instructions in the following form as register mode operations.

OPR Rn

Rn represents a general-purpose register name or number, and OPR represents a general instruction mnemonic. Assembler syntax requires that a general-purpose register be defined as follows.

R0 = %0 (The '%' sign indicates register definition.)
R1 = %1
R2 = %2, etc.

Registers are typically referred to by name as R0, R1, R2, R3, R4, R5, R6, and R7. However, R6 and R7 are also referred to as SP and PC, respectively.

Register Mode Examples (Figures A-7, A-8, and A-9)
(all numbers in octal)

Symbolic	Octal Code	Instruction Name	Operation
INC R3	005203	Increment	One is added to the contents of general-purpose register R3.

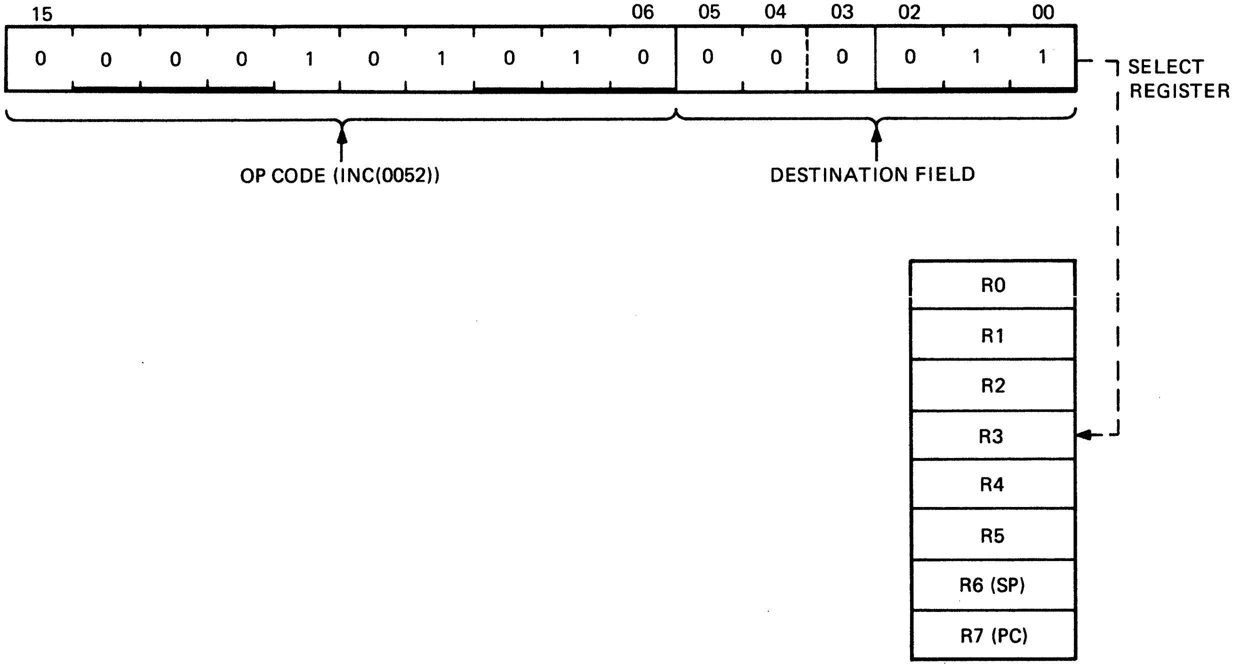


Figure A-7 INC R3

Symbolic	Octal Code	Instruction Name	Operation
ADD R2, R4	060204	Add	The contents of R2 are added to the contents of R4.

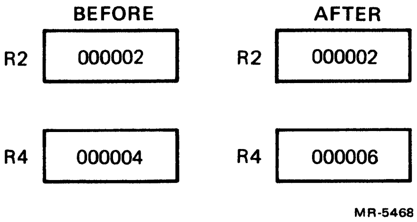


Figure A-8 ADD R2,R4

Symbolic	Octal Code	Instruction Name	Operation
COMB R4	105104	Complement byte	Complement bits 0–7 (byte) of one in R4. (When general-purpose registers are used, byte instructions only operate on bits 0–7; i.e., byte 0 of the register.)

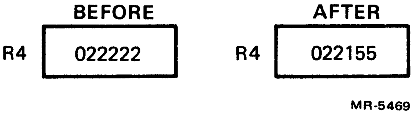


Figure A-9 COMB R4

A.2.3.2 Autoincrement Mode (Mode 2) – Autoincrement mode allows automatic stepping of a pointer through sequential elements of a table of operands. It assumes that the content of the selected general-purpose register is the address of the operand. Contents of registers are stepped (by one for bytes, by two for words, and by two for R6 and R7) to address the next sequential location. The autoincrement mode is especially useful for array processing and stack processing; it accesses an element of a table and then steps the pointer to address the next operand in the table. Although most useful for table handling, this mode is general and may be used for a variety of purposes. The assembler interprets and assembles instructions in the following form as autoincrement mode operations.

OPR (Rn)+

Autoincrement Mode Examples (Figures A-10, A-11, and A-12)

Symbolic	Octal Code	Instruction Name	Operation
CLR (R5)+	005025	Clear	The contents of R5 are used as the address of the operand. The selected operand is cleared, and the contents of R5 are then incremented by two.

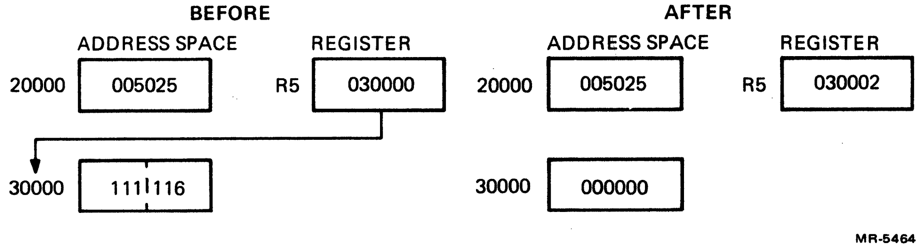


Figure A-10 CLR (R5)+

Symbolic	Octal Code	Instruction Name	Operation
CLRB (R5)+	105025	Clear byte	The contents of R5 are used as the address of the operand. The selected byte operand is cleared, and the contents of R5 are then incremented by one.

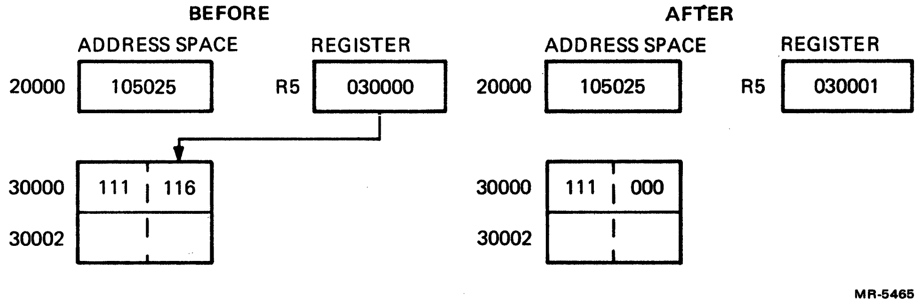


Figure A-11 CLRB (R5)+

Symbolic	Octal Code	Instruction Name	Operation
ADD (R2)+,R4	062204	Add	The contents of R2 are used as the address of the operand that is added to the contents of R4. R2 is then incremented by two.

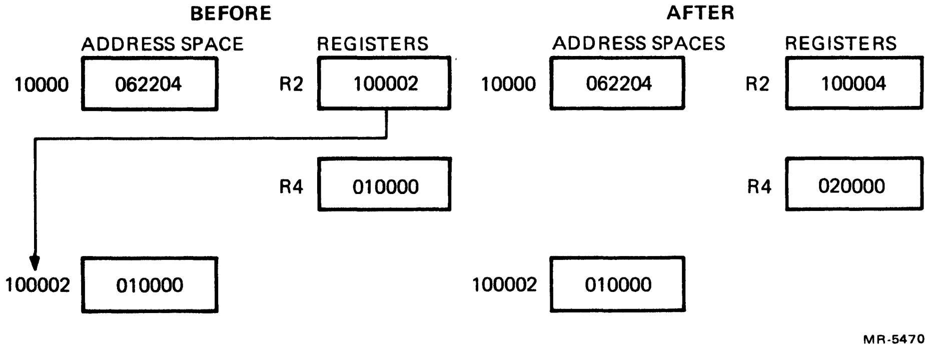


Figure A-12 ADD (R2)+,R4

A.2.3.3 Autodecrement Mode (Mode 4) – Autodecrement mode is useful for processing data in a list in reverse direction. The contents of the selected general-purpose register are decremented (by two for word instructions, by one for byte instructions) and then used as the address of the operand. The choice of postincrement, predecrement features for the KXT11-CA are not arbitrary; they are intended to facilitate hardware/software stack operations. The assembler interprets and assembles instructions in the following form as autodecrement mode operations.

OPR –(Rn)

Autodecrement Mode Examples (Figures A-13, A-14, and A-15)

Symbolic	Octal Code	Instruction Name	Operation
INC –(R0)	005240	Increment	The contents of R0 are decremented by two and used as the address of the operand. The operand is incremented by one.

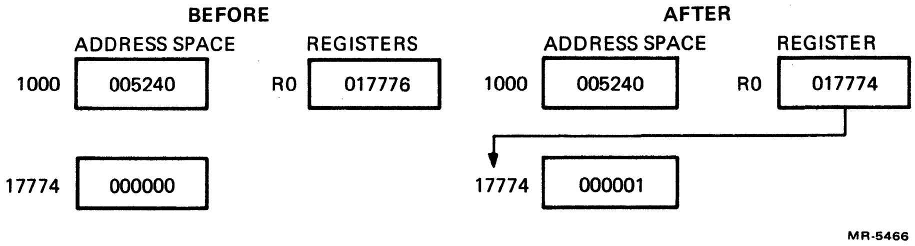


Figure A-13 INC –(R0)

Symbolic	Octal Code	Instruction Name	Operation
INCB –(R0)	105240	Increment byte	The contents of R0 are decremented by one and used as the operand address. The operand byte is increased by one.

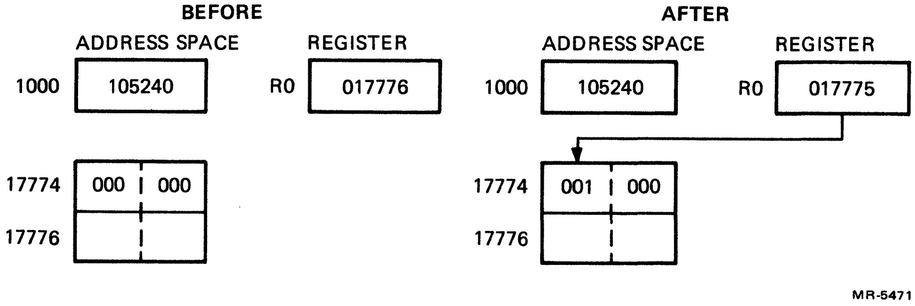


Figure A-14 INCB –(R0)

Symbolic	Octal Code	Instruction Name	Operation
ADD –(R3),R0	064300	Add	The contents of R3 are decremented by two and then used as a pointer to an operand (source) which is added to the contents of R0 (destination operand).

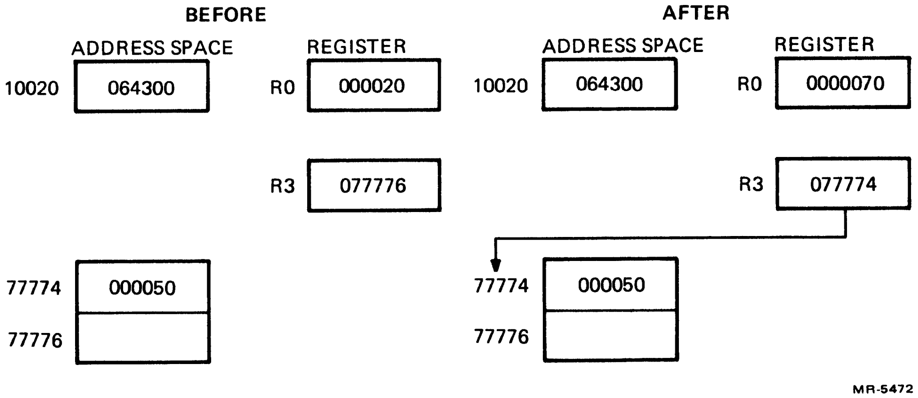


Figure A-15 ADD –(R3),R0

A.2.3.4 Index Mode (Mode 6) – With index mode, the contents of the selected general-purpose register and an index word following the instruction word are summed to form the address of the operand. The contents of the selected register may be used as a base for calculating a series of addresses, thus allowing random access to elements of data structures. The selected register can then be modified by the program to access data in the table. Index addressing instructions are in the following form:

OPR X(Rn)

where X is the indexed word and is located in the memory location following the instruction word, and Rn is the selected general-purpose register.

Index Mode Examples (Figures A-16, A-17, and A-18)

Symbolic	Octal Code	Instruction Name	Operation
CLR 200(R4)	005064 000200	Clear	The address of the operand is determined by adding 200 to the contents of R4. The operand location is cleared.

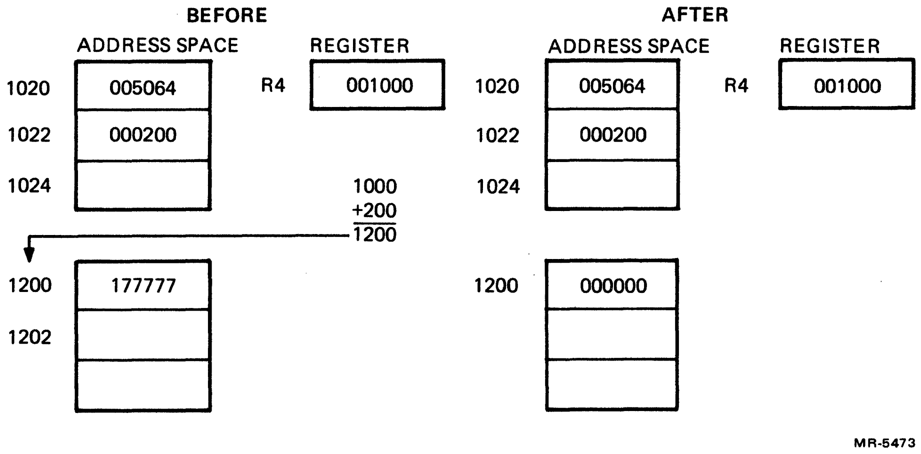


Figure A-16 CLR 200(R4)

Symbolic	Octal Code	Instruction Name	Operation
COMB 200(R1)	105161	Complement byte location	The contents of a 000200 that is determined by adding 200 to the contents of R1 are one's complemented (i.e., logically complemented).

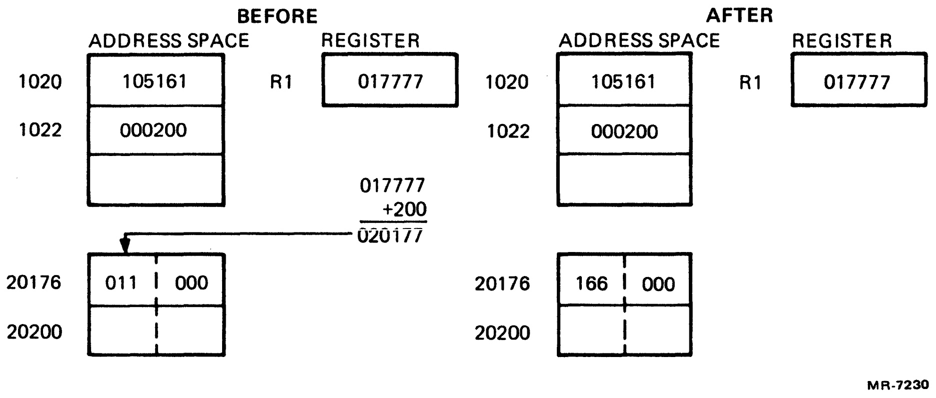


Figure A-17 COMB 200(R1)

Symbolic	Octal Code	Instruction Name	Operation
ADD 30(R2),20(R5)	066265 000030 000020	Add	The contents of a location that is determined by adding 30 to the contents of R2 are added to the contents of a location that is determined by adding 20 to the contents of R5. The result is stored at the destination address, i.e., 20(R5).

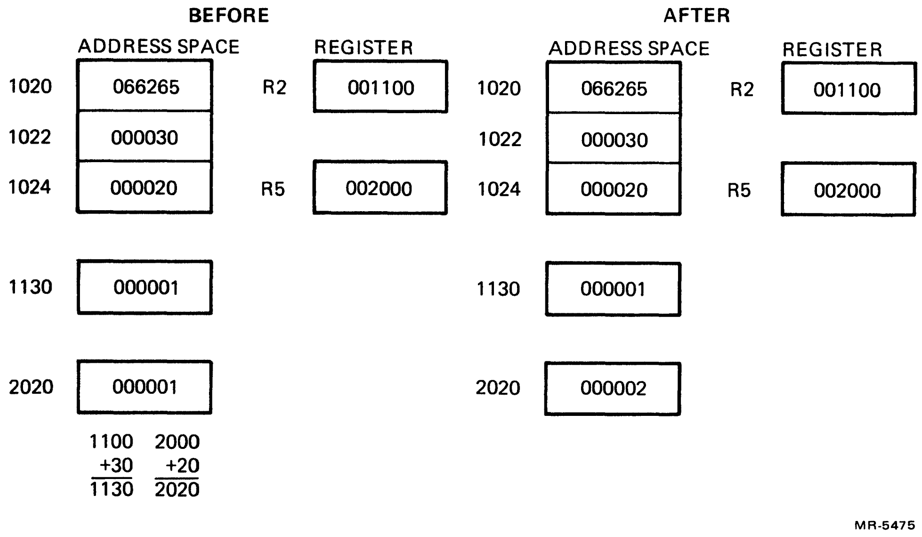


Figure A-18 ADD 30(R2),20(R5)

A.2.4 Deferred (Indirect) Addressing

The four basic modes may also be used with deferred addressing. In the register mode, the operand is the content of the selected register; in the register deferred mode, the content of the selected register is the address of the operand. In the three other deferred modes, the contents of the register select the address of the operand rather than the operand itself. Therefore, these modes are used when a table consists of addresses rather than operands. Assembler syntax for indicating deferred addressing is '@' (or '()' when this is not ambiguous). Table A-3 summarizes the deferred versions of the basic modes. Figures A-19, A-20, A-21, and A-22, which follow the table, illustrate these deferred versions of the basic modes.

Table A-3 Indirect Addressing Modes

Mode	Name	Assembler Syntax	Function
1	Register deferred	@Rn or (Rn)	Register contains the address of the operand.
3	Autoincrement deferred	@(Rn)+	Register is first used as a pointer to a word containing the address of the operand and then incremented (always by two, even for byte instructions).
5	Autodecrement deferred	@–(Rn)	Register is decremented (always by two, even for byte instructions) and then used as a pointer to a word containing the address of the operand.
7	Index deferred	@X(Rn)	Value X (stored in a word following the instruction) and (Rn) are added, and the sum is used as a pointer to a word containing the address of the operand. Neither X nor (Rn) is modified.

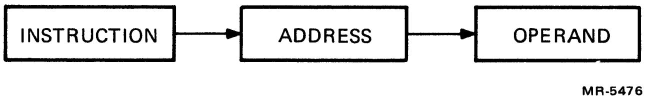


Figure A-19 Mode 1 Register Deferred

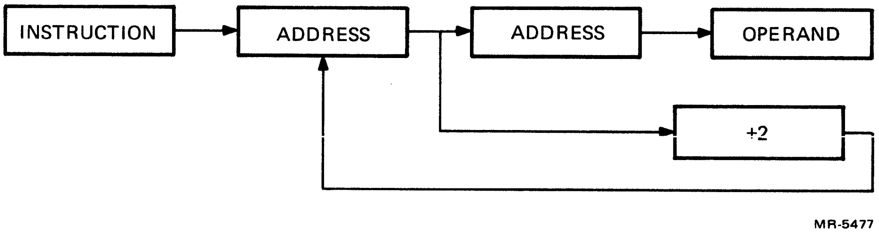


Figure A-20 Mode 3 Autoincrement Deferred

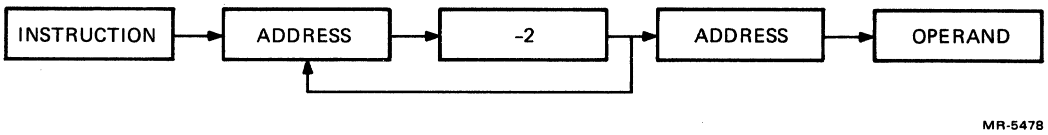


Figure A-21 Mode 5 Autodecrement Deferred

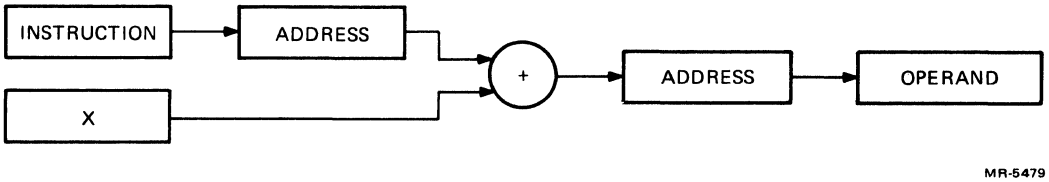


Figure A-22 Mode 7 Index Deferred

Register Deferred Mode Example – Mode 1 (Figure A-23)

Symbolic	Octal Code	Instruction Name	Operation
CLR @R5	005015	Clear	The contents of the location specified in R5 are cleared.

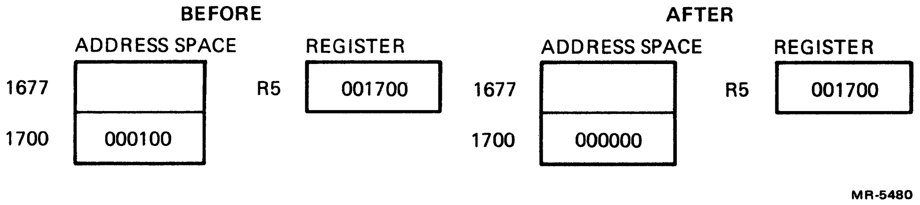


Figure A-23 CLR @R5

Autoincrement Deferred Mode Example – Mode 3 (Figure A-24)

Symbolic	Octal Code	Instruction Name	Operation
INC @(R2)+	005232	Increment	The contents of R2 are used as the address of the address of the operand. The operand is increased by one; and the contents of R2 are incremented by two.

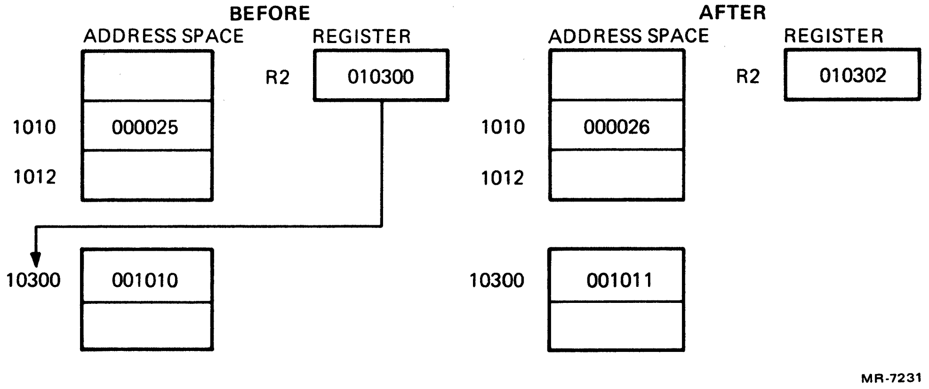


Figure A-24 INC @(R2)+

Autodecrement Deferred Mode Example – Mode 5 (Figure A-25)

Symbolic	Octal Code	Instruction Name	Operation
COM @–(R0)	005150	Complement	The contents of R0 are decremented by two and then used as the address of the address of the operand. Operand is one's complemented (i.e., logically complemented).

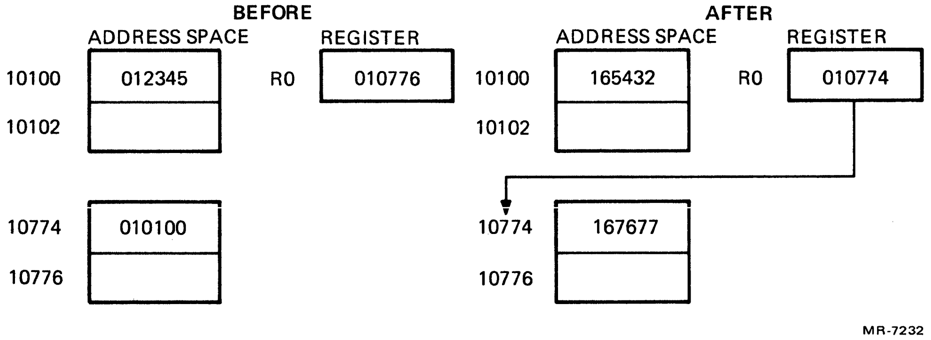


Figure A-25 COM @ –(R0)

Index Deferred Mode Example – Mode 7 (Figure A-26)

Symbolic	Octal Code	Instruction Name	Operation
ADD @1000(R2),R1	067201	Add contents	1000 and the 001000 of R2 are summed to produce the address of the address of the source operand. The contents of the source operand are added to the contents of R1; the result is stored in R1.

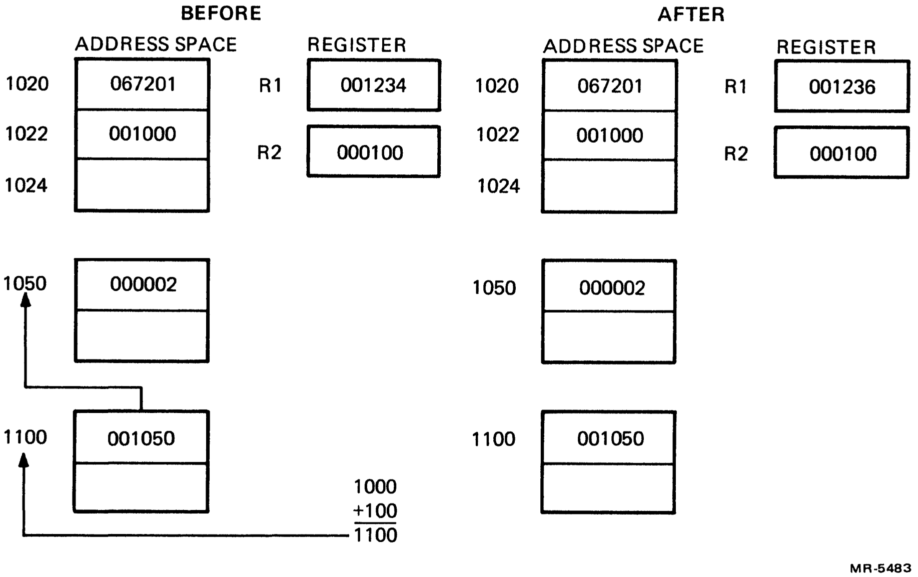


Figure A-26 ADD @1000(R2),R1

A.2.5 Use of the PC as a General-Purpose Register

Although R7 is a general-purpose register, it doubles as the program counter for the microprocessor. Whenever the processor uses the program counter to acquire a word from memory, the program counter is automatically incremented by two to contain the address of the next word of the instruction being executed or the address of the next instruction to be executed. (When the program uses the PC to locate byte data, the PC is still incremented by two.)

The PC responds to all standard KXT11-CA addressing modes. The PC, however, provides advantages for handling position independent code and unstructured data with four of these modes. When utilizing the PC, these modes are termed immediate, absolute (or immediate deferred), relative, and relative deferred. Table A-4 provides a summary of these modes.

Table A-4 PC Register Addressing Modes

Mode	Name	Assembler Syntax	Function
2	Immediate	#n	Operand follows the instruction.
3	Absolute	@#A	Absolute address of operand follows the instruction.
6	Relative	A	Relative address (index value) follows the instruction.
7	Relative deferred	@A	Index value (stored in the word following the instruction) is the relative address for the address of the operand.

When a standard program is available to different users, the ability to load it into different areas of memory and run it is useful. The KXT11-CA can relocate a program efficiently using position independent code (PIC) that is written using the PC addressing modes. If an instruction and its operands are moved so that the relative distance between them is not altered, the same offset relative to the PC can be used in all positions in memory. Thus, PIC usually references locations relative to the current location.

The PC also facilitates the handling of unstructured data. This is particularly true of the immediate and relative modes.

A.2.5.1 Immediate Mode – Using the immediate mode is equivalent to using the autoincrement mode with the PC. It provides time improvements for accessing constant operands by including the constant in the memory location immediately following the instruction word. The assembler interprets and assembles instructions in the following form as immediate mode operations.

```
OPR #n,DD
```

Immediate Mode Example (Figure A-27)

Symbolic	Octal Code	Instruction Name	Operation
ADD #10,R0	062700 000010	Add	The value 10 is located in the second word of the instruction and is added to the contents of R0. Just before this instruction is fetched and executed, the PC points to the first word of the instruction. The processor fetches the first word and increments the PC by two. The source operand mode is 27 (autoincrement the PC). Thus, the PC is used as a pointer to fetch the operand (the second word of the instruction) before being incremented by two to point to the next instruction.

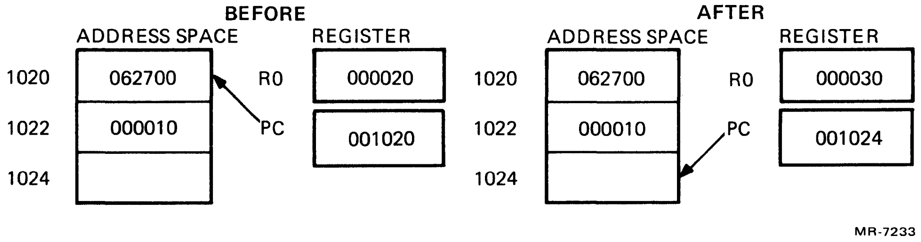


Figure A-27 ADD #10,R0

A.2.5.2 Absolute Addressing – Using the absolute addressing mode is the equivalent of using the immediate deferred or autoincrement deferred modes with the PC. The contents of the location following the instruction are taken as the address of the operand. Immediate data is interpreted as an absolute address (i.e., an address that remains constant no matter where in memory the assembled instruction is executed). The assembler interprets and assembles instructions in the following form as absolute addressing mode operations.

OPR @#A

Absolute Mode Examples (Figures A-28 and A-29)

Symbolic	Octal Code	Instruction Name	Operation
CLR @#1100	005037 001100	Clear	The the contents of location 1100 are cleared.

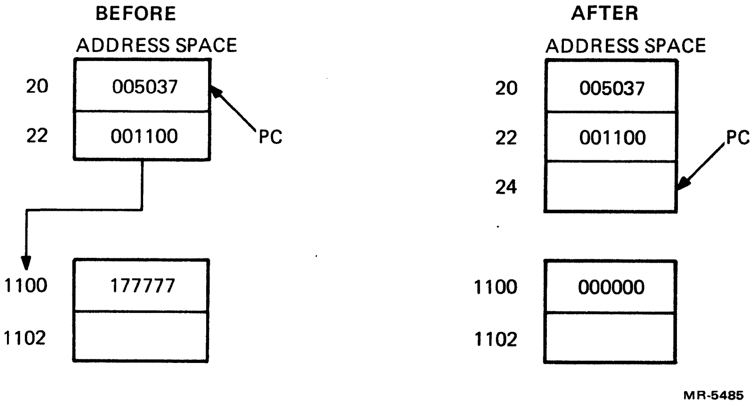


Figure A-28 CLR @#1100

Symbolic	Octal Code	Instruction Name	Operation
ADD @#2000,R3	063703 002000	Add	The contents of location 2000 are added to R3.

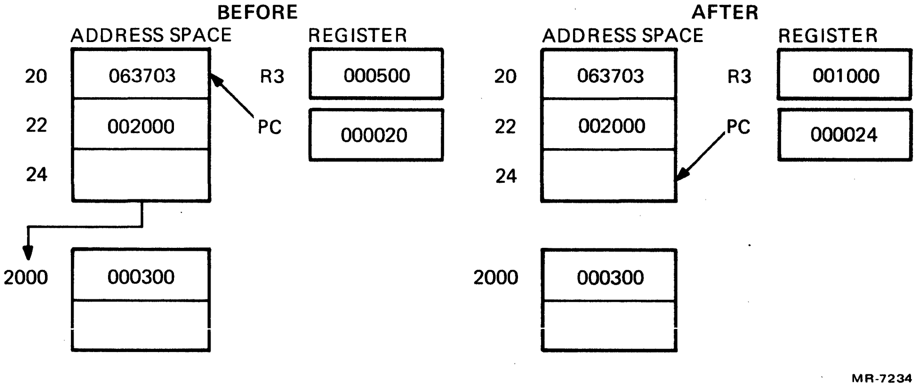


Figure A-29 ADD @#2000,R3

A.2.5.3 Relative Addressing – The relative addressing mode is assembled as index mode using R7. The base of the address calculation, which is stored in the second or third word of the instruction, is not the address of the operand, but the number that, when added to the PC, becomes the address of the operand. This mode is useful for writing position independent code because the location referenced is always fixed relative to the PC. When instructions are to be relocated, the operand is moved by the same amount. The assembler interprets and assembles instructions in the following forms as relative addressing mode operations.

OPR A
or
OPR X(PC)

where X is the location of A relative to the instruction.

Relative Addressing Example (Figure A-30)

Symbolic	**Octal Code**	**Instruction Name**	**Operation**
INC A	005267	Increment	
	000054		To increment location A, contents of memory location immediately following instruction word are added to (PC) to produce address A. Contents of A are increased by one.

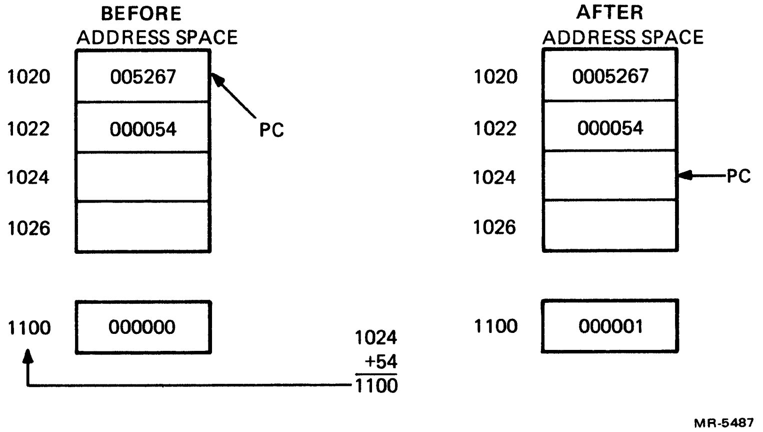


Figure A-30 INC A

A.2.5.4 Relative Deferred Addressing – The relative deferred addressing mode is similar to the relative mode. The second word of the instruction, however, when added to the PC, contains the address of the address of the operand rather than the address of the operand. The assembler interprets and assembles instructions in the following forms as relative deferred addressing mode operations.

OPR @A
or
OPR @X(PC)

where X is the location containing the address of A relative to the instruction.

Relative Deferred Mode Example (Figure A-31)

Symbolic	**Octal Code**	**Instruction Name**	**Operation**
CLR @A	005077	Clear instruction	The second word of 000020 is added to updated PC to produce the address of the address of the operand. The operand is cleared.

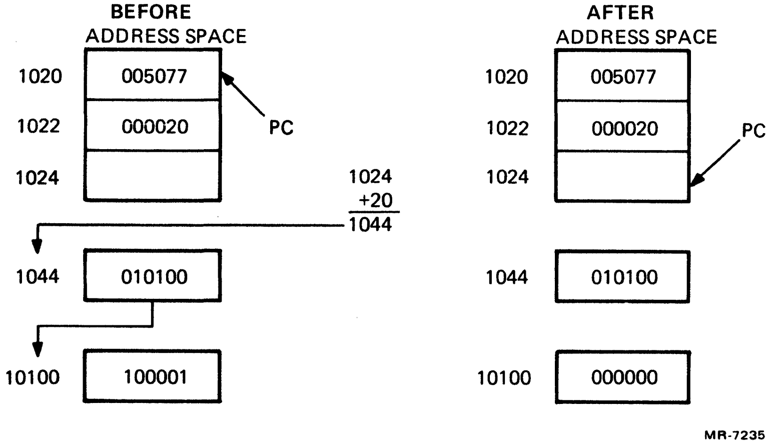


Figure A-31 CLR @A

A.2.6 Use of the Stack Pointer as a General-Purpose Register

The processor stack pointer (SP, register R6) is the general-purpose register most often used for stack operations related to program nesting. Autodecrement with register R6 pushes data onto the stack, and autoincrement with register R6 pops data off the stack. Since the SP is used by the processor for interrupt handling, it has a special attribute: autoincrements and autodecrements are always done in steps of two. Byte operations using the SP in this way leave odd addresses unmodified.

A.3 INSTRUCTION SET

Specifications for each instruction in the KXT11-CA instruction set follow and include each instruction's mnemonic, octal code, binary code, a diagram showing the format of the instruction, a symbolic notation describing its execution and effect on the condition codes, a description, special comments, and examples.

MNEMONIC: A mnemonic is indicated before each description. When the word instruction has a byte equivalent, the byte mnemonic is also shown.

INSTRUCTION FORMAT: A diagram accompanying each instruction shows the octal op code, binary op code, and bit assignments. In byte instructions, the most significant bit (bit 15) is always a one.

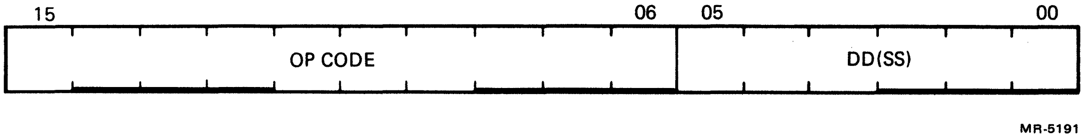
SYMBOLS: The following symbols are used in the instruction specifications.

()	= contents of
SS or src	= source address
DD or dst	= destination address
loc	= location
←	= becomes
↑	= "is popped from stack"
↓	= "is pushed onto stack"
∧	= boolean AND
∨	= boolean OR
∀	= exclusive OR
~	= boolean not
Reg or R	= register
B	= Byte
■	0 for word = 1 for byte
,	= concatenated

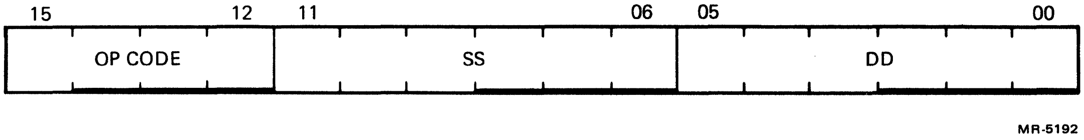
A.3.1 Instruction Formats

The following formats include all instructions used in the KXT11-CA SBC. Refer to individual instructions for more detailed information.

1. Single-operand group: CLR, CLRB, COM, COMB, INC, INCB, DEC, DECB, NEG, NEGB, ADC, ADCB, SBC, SBCB, TST, TSTB, ROR, RORB, ROL, ROLB, ASR, ASRB, ASL, ASLB, JMP, SWAB, MFPS, MTPS, SXT, XOR

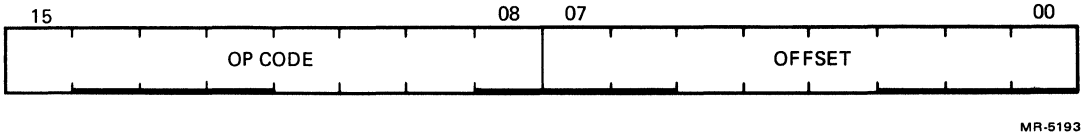


2. Double-operand group: BIT, BITB, BIC, BICB, BIS, BISB, ADD, SUB, MOV, MOVB, CMP, CMPB

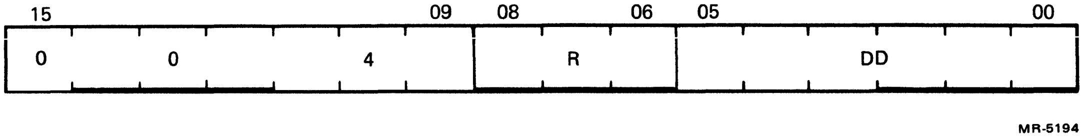


3. Program control group:

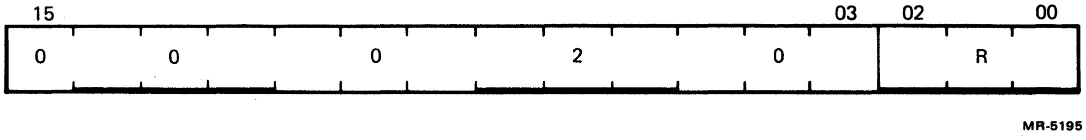
   a. Branch (all branch instructions)



   b. Jump to subroutine (JSR)



   c. Subroutine return (RTS)



d. Traps (breakpoint, IOT, EMT, TRAP, BPT)

15		00
	OP CODE	

MR-5196

e. Subtract 1 and branch if = 0 (SOB)

15		09	08	06	05	00
0	0	7	R		NN	

MR-5197

4. Operate group: HALT, WAIT, RTI, RESET, RTT, NOP, MFPT

15		00
	OP CODE	

MR-5198

5. Condition code operators: (all condition code instructions)

15			06	05	04	03	02	01	00
0	0	0	2	4	0/1	N	Z	V	C

MR-5199

Byte Instructions – The KXT11-CA includes a full complement of instructions that manipulate byte operands. Because all microprocessor addressing is byte-oriented, byte manipulation addressing is straightforward. Byte instructions with autoincrement or autodecrement direct addressing cause the specified register to be modified by one to point to the next byte of data. Byte operations in register mode access the low-order byte of the specified register. These provisions enable the KXT11-CA SBC to perform as either a word or byte microprocessor. The numbering scheme for word and byte addresses in memory is illustrated in Figure A-32.

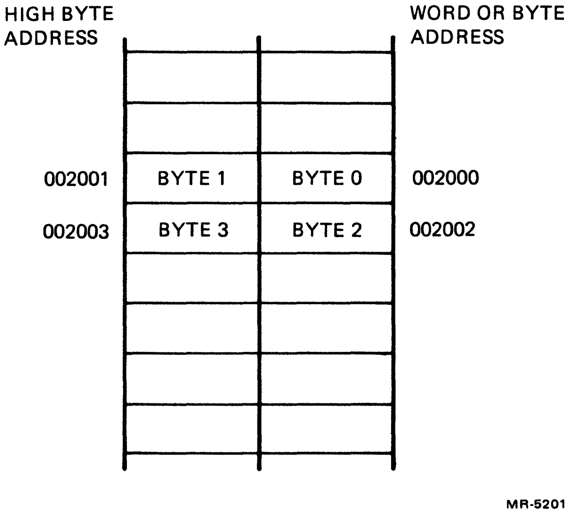


Figure A-32 Byte Instructions

The most significant bit (bit 15) of the instruction word is set to indicate a byte instruction.

Byte Instruction Example

Symbolic	Octal Code	Instruction Name
CLR	0050DD	Clear word
CLRB	1050DD	Clear byte

A.3.2 List of Instructions

The KXT11-CA instruction set is shown in Table A-5.

Table A-5 KXT11-CA Instruction Set

Mnemonic	Instruction	Op Code
SINGLE OPERAND		
General		
CLR(B)	Clear dst	■050DD
COM(B)	Complement dst	■051DD
INC(B)	Increment dst	■052DD
DEC(B)	Decrement dst	■053DD
NEG(B)	Negate dst	■054DD
TST(B)	Test dst	■057DD
Shift and Rotate		
ASR(B)	Arithmetic shift right	■062DD
ASL(B)	Arithmetic shift left	■063DD
ROR(B)	Rotate right	■060DD
ROL(B)	Rotate left	■061DD
SWAB	Swap bytes	0003DD
Multiple Precision		
ADC(B)	Add carry	■055DD
SBC(B)	Subtract carry	■056DD
SXT	Sign extend	0067DD
PS Word Operators		
MFPS	Move byte from PS	1067DD
MTPS	Move byte to PS	1064SS
DOUBLE OPERAND		
General		
MOV(B)	Move source to destination	■1SSDD
CMP(B)	Compare src to dst	■2SSDD
ADD	Add src to dst	06SSDD
SUB	Subtract src from dst	16SSDD
Logical		
BIT(B)	Bit test	■3SSDD
BIC(B)	Bit clear	■4SSDD
BIS(B)	Bit set	■5SSDD
XOR	Exclusive OR	074RDD

Table A-5 KXT11-CA Instruction Set (Cont)

Mnemonic	Instruction	Op Code
PROGRAM CONTROL		
Branch		
BR	Branch (unconditional)	000400
BNE	Branch if not equal (to zero)	001000
BEQ	Branch if equal (to zero)	001400
BPL	Branch if plus	100000
BMI	Branch if minus	100400
BVC	Branch if overflow is clear	102000
BVS	Branch if overflow is set	102400
BCC	Branch if carry is clear	103000
BCS	Branch if carry is set	103400
Signed Conditional Branch		
BGE	Branch if greater than or equal (to zero)	002000
BLT	Branch if less than (zero)	002400
BGT	Branch if greater than (zero)	003000
BLE	Branch if less than or equal (to zero)	003400
Unsigned Conditional Branch		
BHI	Branch if higher	101000
BLOS	Branch if lower or same	101400
BHIS	Branch if higher or same	103000
BLO	Branch if lower	103400
Jump and Subroutine		
JMP	Jump	0001DD
JSR	Jump to subroutine	004RDD
RTS	Return from subroutine	00020R
SOB	Subtract one and branch (if ≠ 0)	077R00
Trap and Interrupt		
EMT	Emulator trap	104000–104377
TRAP	Trap	104400–104777
BPT	Breakpoint trap	000003
IOT	Input/output trap	000004
RTI	Return from interrupt	000002
RTT	Return from interrupt	000006
MISCELLANEOUS		
HALT	Halt	000000
WAIT	Wait for interrupt	000001
RESET	Reset external bus	000005
MFPT	Move processor type	000007

Table A-5 KXT11-CA Instruction Set (Cont)

Mnemonic	Instruction	Op Code
RESERVED INSTRUCTIONS		
		00021R 00022R
CONDITION CODE OPERATORS		
CLC	Clear C	000241
CLV	Clear V	000242
CLZ	Clear Z	000244
CLN	Clear N	000250
CCC	Clear all CC bits	000257
SEC	Set C	000261
SEV	Set V	000262
SEZ	Set Z	000264
SEN	Set N	000270
SCC	Set all CC bits	000277
NOP	No operation	000240

A.3.3 Single-Operand Instructions

NOTE

In most KXT11-CA instructions, a write operation to a memory location or register is always preceded by a read operation from the same location except when writing PC and processor status (PS) to the stack in the following two cases.

1. **The execution of the microcode preceding an interrupt or trap service routine.**

2. **Interrupt and trap instructions:**

   HLT
   TRAP
   BPT
   IOT

A.3.3.1 General -

CLR
CLRB

Clear Destination ■050DD

15									06	05					00
0/1	0	0	0	1	0	1	0	0	0	d	d	d	d	d	d

MR-5202

Operation: (dst) ← 0

Condition Codes:

N: cleared
Z: set
V: cleared
C: cleared

Description:

Word: Contents of specified destination are replaced with zeros.
Byte: Same

Example:

CLR R1

Before	After
(R1) = 177777	(R1) = 000000
N Z V C	N Z V C
1 1 1 1	0 1 0 0

COM
COMB

Complement Destination ■051DD

15									06	05					00
0/1	0	0	0	1	0	1	0	0	1	d	d	d	d	d	d

MR-5203

Operation: (dst) ← ~ (dst)

Condition Codes:

N: set if most significant bit of result is set; cleared otherwise
Z: set if result = 0; cleared otherwise
V: cleared
C: set

Description: Word: The contents of the destination address are replaced by their logical complement (each bit equal to zero is set, and each bit equal to one is cleared).

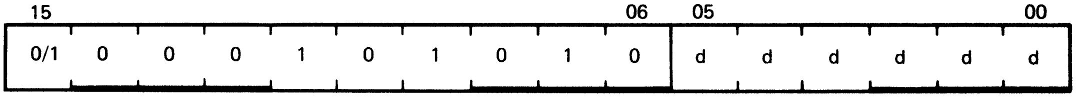
Byte: Same

Example: COM R0

Before	After
(R0) = 013333	(R0) = 164444
N Z V C	N Z V C
0 1 1 0	1 0 0 1

INC
INCB

Increment Destination ■052DD



15									06	05					00
0/1	0	0	0	1	0	1	0	1	0	d	d	d	d	d	d

MR-5204

Operation: (dst) ← (dst) + 1

Condition Codes:

N: set if result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: set if (dst) held 077777; cleared otherwise
C: not affected

Description: Word: One is added to contents of destination.

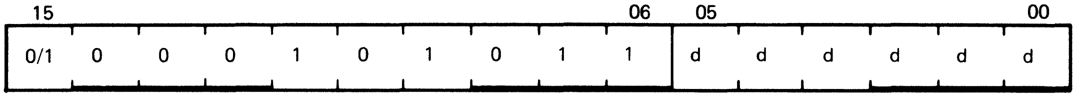
Byte: Same

Example: INC R2

Before	After
(R2) = 000333	(R2) = 000334
N Z V C	N Z V C
0 0 0 0	0 0 0 0

DEC
DECB

Decrement Destination ■053DD



MR-5205

Operation: (dst) ← (dst) −1

Condition Codes:
N: set if result < 0, cleared otherwise
Z: set is result = 0; cleared otherwise
V: set if (dst) was 100000; cleared otherwise
C: not affected

Description:
Word: One is subtracted from the contents of the destination.
Byte: Same

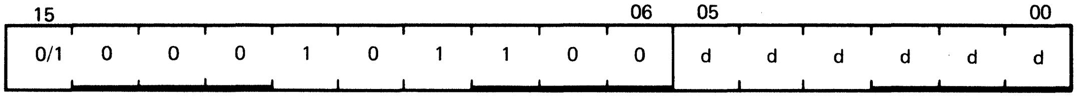
Example:

DEC R5

Before	After
(R5) = 000001	(R5) = 000000
N Z V C	N Z V C
1 0 0 0	0 1 0 0

NEG
NEGB

Negate Destination ■054DD



MR-5206

Operation: (dst) ← − (dst)

Condition Codes:
N: set if the result < 0; cleared otherwise
Z: set if the result = 0; cleared otherwise
V: ~~set if the result is 100000; cleared otherwise~~
C: cleared if the result is 0; set otherwise

Description:
Word: The contents of the destination address are replaced by its two's complement. 100000 is replaced by itself (in two's complement notation, the most negative number has no positive counterpart).
Byte: Same

Example:

NEG R0

Before	After
(R0) = 000010	(R0) = 177770
N Z V C	N Z V C
0 0 0 0	1 0 0 1

TST
TSTB

Test Destination ■057DD

15									06	05					00
0/1	0	0	0	1	0	1	1	1	1	d	d	d	d	d	d

MR-5207

Operation: (dst) ← (dst)

Condition Codes:

N: set if the result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: cleared
C: cleared

Description:

Word: The condition codes N and Z are set according to the contents of the destination address, and the contents of the destination remain unmodified.
Byte: Same

Example:

TST R1

Before	After
(R1) = 012340	(R1) = 012340
N Z V C	N Z V C
0 0 1 1	0 0 0 0

A.3.3.2 Shifts and Rotates – Scaling data by factors of two is accomplished with two shift instructions:

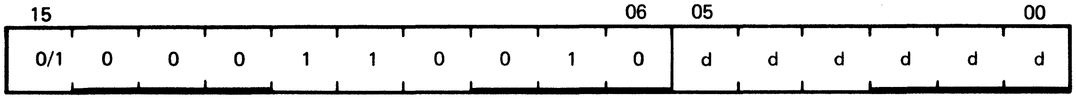
1. ASR – Arithmetic shift right
2. ASL – Arithmetic shift left

The sign bit (bit 15) of the operand is reproduced in shifts to the right. The low-order bit is filled with zero in shifts to the left. Bits shifted out of the C-bit, as shown in the following examples, are lost.

The rotate instructions operate on the destination word and the C-bit as though they formed a 17-bit circular buffer. These instructions facilitate sequential bit testing and detailed bit manipulation.

ASR
ASRB

Arithmetic Shift Right ■062DD



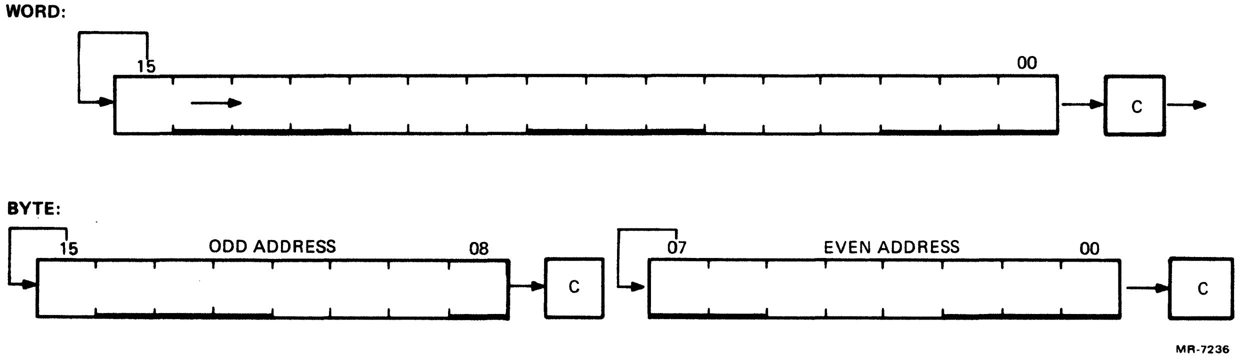
15									06	05					00
0/1	0	0	0	1	1	0	0	1	0	d	d	d	d	d	d

Operation: (dst) ← (dst) shifted one place to the right

Condition Codes:
N: set if the high-order bit of the result is set (result < 0); cleared otherwise
Z: set if the result = 0; cleared otherwise
V: loaded from the exclusive OR of the N-bit and C-bit (as set by the completion of the shift operation)
C: loaded from the low-order bit of the destination

Description:
Word: All bits of the destination are shifted right one place. Bit 15 is reproduced. The C-bit is loaded from bit 0 of the destination. ASR performs signed division of the destination by two.
Byte: Same

Example:



ASL
ASLB

Arithmetic Shift Left ■063DD

15									06	05					00
0/1	0	0	0	1	1	0	0	1	1	d	d	d	d	d	d

MR-5210

Operation: (dst) ← (dst) shifted one place to the left

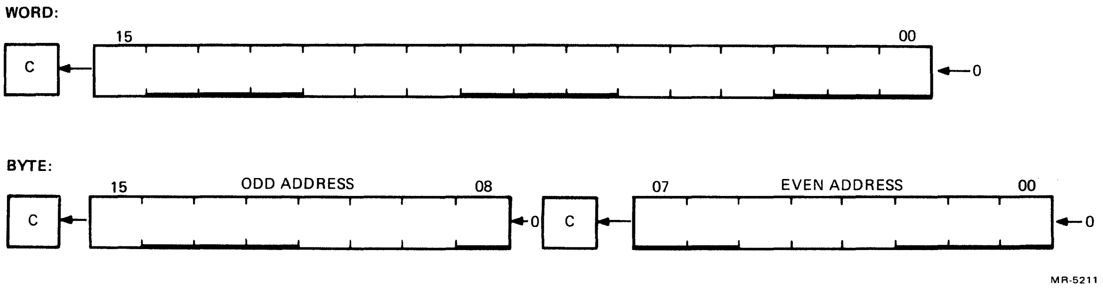
Condition Codes:

N: set if the high-order bit of the result is set (result < 0); cleared otherwise
Z: set if the result = 0; cleared otherwise
V: loaded with the exclusive OR of the N-bit and C-bit (as set by the completion of the shift operation)
C: loaded with the high-order bit of the destination

Description:

Word: All bits of the destination are shifted left one place. Bit 0 is loaded with a zero. The C-bit of the status word is loaded from the most significant bit of the destination. ASL performs signed multiplication of the destination by two with overflow indication.
Byte: Same

Example:



ROR
RORB

Rotate Right

■060DD

15									06	05					00
0/1	0	0	0	1	1	0	0	0	0	d	d	d	d	d	d

MR-5212

Operation: (dst) ← (dst) rotate right one place

Condition Codes:

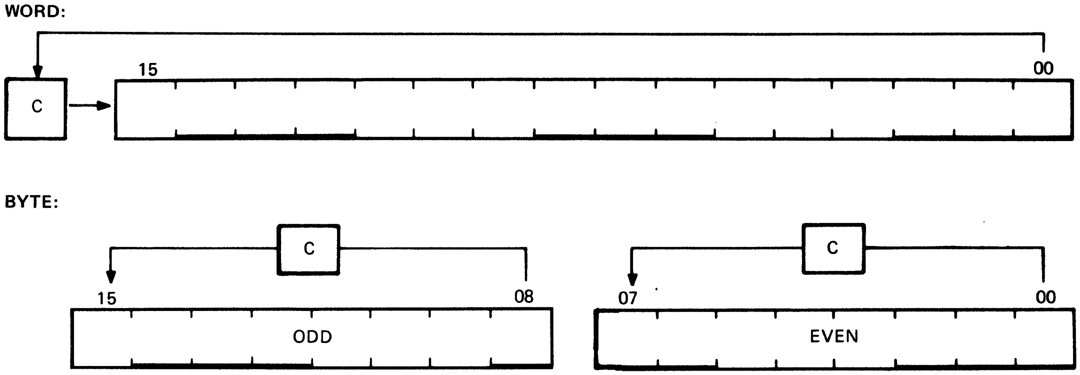
N: set if the high-order bit of the result is set (result < 0); cleared otherwise
Z: set if all bits of result = 0; cleared otherwise
V: loaded with the exclusive OR of the N-bit and C-bit (as set by the completion of the rotate operation)
C: loaded with the low-order bit of the destination

Description:

Word: All bits of the destination are rotated right one place. Bit 0 is loaded into the C-bit, and the previous contents of the C-bit are loaded into bit 15 of the destination.

Byte: Same

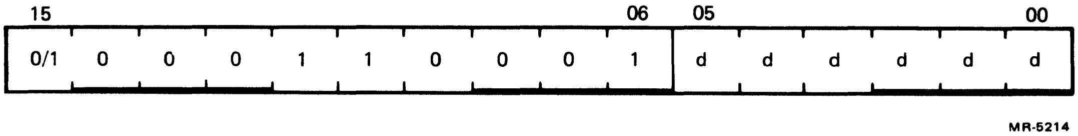
Example:



MR-5213

ROL
ROLB

Rotate Left ■061DD



Operation: (dst) ← (dst) rotate left one place

Condition Codes:

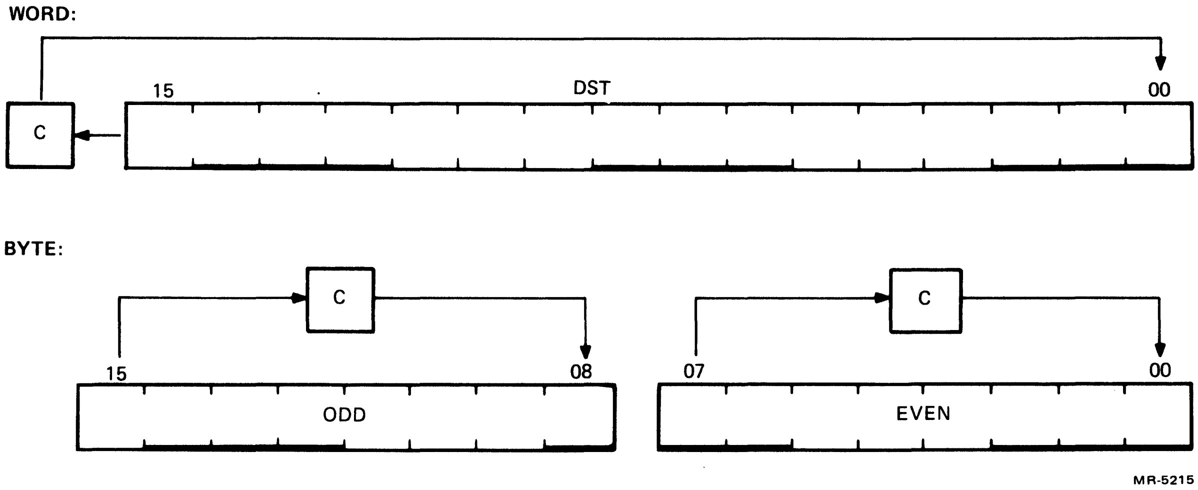
N: set if the high-order bit of the result word is set (result < 0); cleared otherwise
Z: set if all bits of the result word = 0; cleared otherwise
V: loaded with the exclusive OR of the N-bit and C-bit (as set by the completion of the rotate operation)
C: loaded with the high-order bit of the destination

Description:

Word: All bits of the destination are rotated left one place. Bit 15 is loaded into the C-bit of the status word, and the previous contents of the C-bit are loaded into bit 0 of the destination.

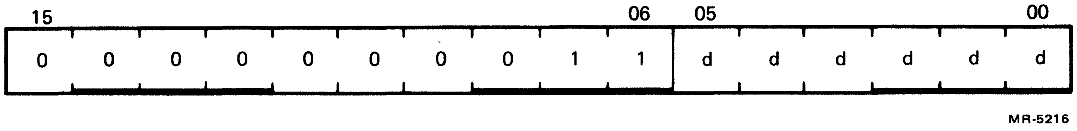
Byte: Same

Example:



SWAB

Swap Bytes 0003DD



Operation: Byte 1/Byte 0 ← Byte 0/Byte 1

Condition Codes:

N: set if the high-order bit of the low-order byte (bit 7) of the result is set; cleared otherwise
Z: set if low-order byte of result = 0; cleared otherwise
V: cleared
C: cleared

Description: High-order byte and low-order byte of the destination word are exchanged (destination must be a word address).

Example:

SWAB R1

Before	After
(R1) = 077777	(R1) = 177577
N Z V C	N Z V C
1 1 1 1	0 0 0 0

A.3.3.3 Multiple Precision – It is sometimes necessary to do arithmetic on operands considered as multiple words or bytes. The SBC-11/21 PLUS makes special provisions for such operations with the instructions ADC (add carry) and SBC (subtract carry) and their byte equivalents. For example, two 16-bit words may be combined into a 32-bit double precision word and added or subtracted as shown in Figure A-33.

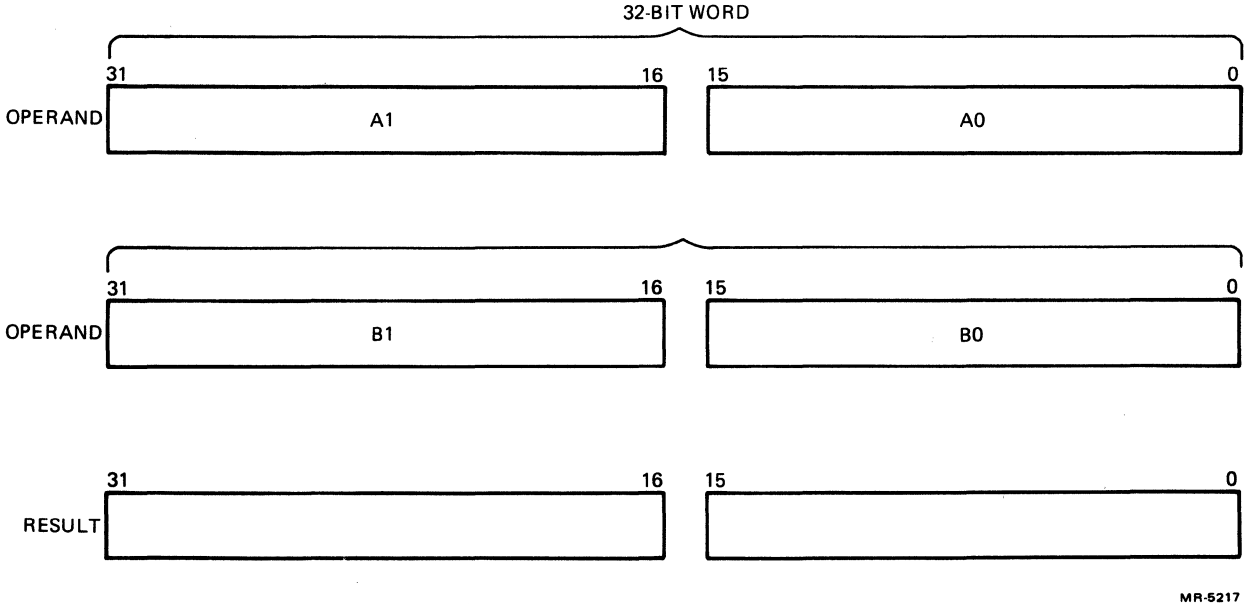


Figure A-33 Multiple Precision

Multiple Precision Example

The addition of −1 and −1 could be performed as follows:

−1 = 37777777777

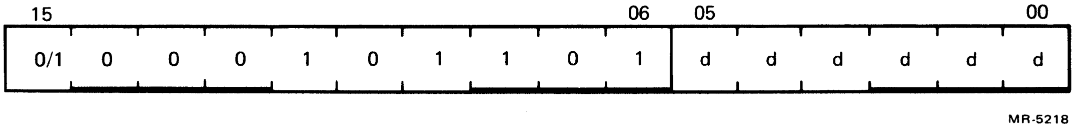
(R1) = 177777 (R2) = 177777 (R3) = 177777 (R4) = 177777

```
ADD R1,R2
ADC R3
ADD R4,R3
```

1. After (R1) and (R2) are added, 1 is loaded into the C-bit
2. ADC instruction adds C-bit to (R3); (R3) = 0
3. (R3) and (R4) are added
4. Result is 37777777776 or −2

ADC
ADCB

Add Carry ■055DD



Operation: (dst) ← (dst) + (C-bit)

Condition Codes:
N: set if result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: set if (dst) was 077777 and (C) was 1; cleared otherwise
C: set if (dst) was 177777 and (C) was 1; cleared otherwise

Description:
Word: The contents of the C-bit are added into the destination. This permits the carry from the addition of the low-order words to be carried into the high-order result.
Byte: Same

Example: Double precision addition is done with the following instruction sequence:

```
ADD A0,B0        add low-order parts
ADC B1           add carry into high order
ADD A1,B1        add high-order parts
```

SBC
SBCB

Subtract Carry ■056DD

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

MR-5219

Operation: (dst) ← (dst) − (C)

Condition Codes:
N: set if result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: set if (dst) was 100000; cleared otherwise
C: set if (dst) was 0 and C was 1; cleared otherwise

Description:
Word: The contents of the C-bit are subtracted from the destination. This permits the carry from the subtraction of two low-order words to be subtracted from the high-order part of the result.
Byte: Same

Example: Double precision subtraction is done with the following instruction sequence:

```
SUB A0,B0
SBC B1
SUB A1,B1
```

SXT

Sign Extend 0067DD

15									06	05					00
0	0	0	0	1	1	0	1	1	1	d	d	d	d	d	d

MR-5220

Operation:
(dst) ← 0 if N-bit is clear
(dst) ← 1 if N-bit is set

Condition Codes:
N: unaffected
Z: ~~set if N-bit is clear~~
V: cleared
C: unaffected

Description: If the condition code bit N is set, a −1 is placed in the destination operand; if the N-bit is clear, then a zero is placed in the destination operand. This instruction is particularly useful in multiple precision arithmetic because it permits the sign to be extended through multiple words.

Example: SXT A

Before	After
(A) = 012345	(A) = 177777
N Z V C	N Z V C
1 0 0 0	1 0 0 0

A.3.3.4 PS Word Operators -

MFPS

Move Byte from Processor Status (PS) 1067DD

15							08	07							00
1	0	0	0	1	1	0	1	1	1	d	d	d	d	d	d

MR-5221

Operation: (dst) ← PS
dst lower 8 bits

Condition Codes:
N: set if PS bit 7 = 1; cleared otherwise
Z: set if PS <0:7> = 0; cleared otherwise
V: cleared
C: not affected

Description: The 8-bit contents of the PS are moved to the effective destination. If the destination is mode 0, PS bit 7 is sign extended through the upper byte of the register. The destination operand address is treated as a byte address.

Example: MFPS R0

Before	After
R0 [0]	R0 [000014]
PS [000014]	PS [000000]

MTPS

Move Byte to Processor Status 1064SS

15							08	07							00
1	0	0	0	1	1	0	1	0	0	s	s	s	s	s	s

MR-5222

Operation: PS ← (src)

Condition Codes: Set according to effective source operand bits 0–3

Description: The 8 bits of the effective operand replace the current contents of the PS. The source operand address is treated as a byte address. The T-bit (PS bit 4) cannot be set with this instruction. The source operand remains unchanged. This instruction can be used to change the priority bits (PS bits 7–5) in the PS.

A.3.4 Double-Operand Instructions

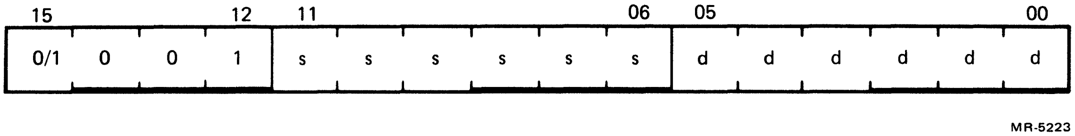
Double operand instructions save instructions and time because they eliminate the need for load and save sequences such as those used in accumulator-oriented machines.

A.3.4.1 General -

MOV
MOVB

Move Source to Destination

■1SSDD



Operation: (dst) ← (src)

Condition Codes:

N: set if (src) < 0; cleared otherwise
Z: set if (src) = 0; cleared otherwise
V: cleared
C: not affected

Description:

Word: The source operand is moved to the destination location. The previous contents of the destination are lost. The contents of the source address are not affected.

Byte: Same as MOV. The MOVB to a register (unique among byte instructions) extends the most significant bit of the low-order byte (sign extension). Otherwise, MOVB operates on bytes exactly as MOV operates on words.

Example:

MOV XXX,R1	loads register 1 with the contents of memory location; XXX represents a programmer-defined mnemonic used to represent a memory location
MOV #20,R0	loads the number 20 into register 0; '#' indicates that the value 20 is the operand
MOV @#20,–(R6)	pushes the operand contained in location 20 onto the stack
MOV (R6)+,@#177566	pops the operand off a stack and moves it into memory location 177566 (terminal print buffer)
MOV R1,R3	performs an inter-register transfer
MOVB @#177562,@#177566	moves a character from terminal keyboard buffer to terminal printer buffer

CMP
CMPB

Compare Source to Destination ■2SSDD

15			12	11					06	05					00
0/1	0	1	0	s	s	s	s	s	s	d	d	d	d	d	d

MR-5224

Operation: (src) − (dst)

Condition Codes:

N: set if result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: set if there was arithmetic overflow; that is, operands were of opposite signs and the sign of the destination was the same as the sign of the result; cleared otherwise
C: cleared if there was a carry from the most significant bit of the result; set otherwise

Description:

Word: The source and destination operands are compared, and the condition codes are set. The condition codes may then be used for arithmetic and logical conditional branches. Both operands are unaffected. The only action is to set the condition codes. The compare is customarily followed by a conditional branch instruction. Unlike the subtract instruction, the order of operation is (src) − (dst), not (dst) − (src).
Byte: Same

ADD

Add Source to Destination 06SSDD

15			12	11					06	05					00
0	1	1	0	s	s	s	s	s	s	d	d	d	d	d	d

MR-5225

Operation: (dst) ← (src) + (dst)

Condition Codes:

N: set if result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: set if there was arithmetic overflow as a result of the operation; that is, both operands were of the same sign and the result was of the opposite sign; cleared otherwise
C: set if there was a carry from the most significant bit of the result; cleared otherwise

Description:

Word: The source operand is added to the destination operand and the result is stored at the destination address. The original contents of the destination are lost. The contents of the source are not affected. Two's complement addition is performed.

Byte: There is no equivalent byte mode.

Example:

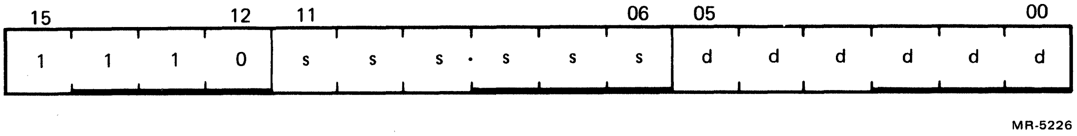
ADD 20,R0	add to register
ADD R1,XXX	add to memory
ADD R1,R2	add register to register
ADD @#17750,XXX	add memory to memory

XXX is a programmer-defined mnemonic for a memory location.

SUB

Subtract Source from Destination

16SSDD



Operation: (dst) ← (dst) − (src)

Condition Codes:

N: set if result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: set if there was arithmetic overflow as a result of the operation, that is if operands were of opposite signs and the sign of the source was the same as the sign of the result; cleared otherwise
C: cleared if there was a carry from the most significant bit of the result; set otherwise

Description:

Word: The source operand is subtracted from the destination operand, and the result is left at the destination address. The original contents of the destination are lost. The contents of the source are not affected. In double-precision arithmetic, the C-bit, when set, indicates a borrow.

Byte: There is no equivalent byte mode.

Example:

SUB R1,R2

Before	After
(R1) = 011111	(R1) = 011111
(R2) = 012345	(R2) = 001234
N Z V C	N Z V C
1 1 1 1	0 0 0 0

A.3.4.2 Logical – Logical group instructions have the same format as the double operand arithmetic group. They permit operations on data at the bit level.

BIT
BITB

Bit Test ■3SSDD

15			12	11					06	05					00
0/1	0	1	1	s	s	s	s	s	s	d	d	d	d	d	d

MR-5227

Operation: (src) ∧ (dst)

Condition Codes:

N: set if high-order bit of result is set; cleared otherwise
Z: set if result = 0; cleared otherwise
V: cleared
C: not affected

Description:

Word: Logical "and" comparison of the source and destination operands is performed, and condition codes are modified accordingly. Neither the source nor destination is affected. The BIT instruction may be used either to test whether any of the corresponding bits that are set in the destination are also set in the source or whether all corresponding bits set in the destination are clear in the source.
Byte: Same

Example:

BIT #30,R3 test bits three and four of R3 to see if both are off

R3 = 0 000 000 000 011 000

Before	After
N Z V C	N Z V C
1 1 1 1	0 0 0 1

BIC
BICB

Bit Clear ■4SSDD

15			12	11					06	05					00
0/1	1	0	0	s	s	s	s	s	s	d	d	d	d	d	d

MR-5228

Operation: (dst) ← (dst) ~ (src)

Condition Codes:

N: set if high-order bit of result is set; cleared otherwise
Z: set if result = 0; cleared otherwise
V: cleared
C: not affected

Description:

Word: Each bit in the destination that corresponds to a set bit in the source is cleared. The original contents of the destination are lost. The contents of the source are not affected.

Byte: Same

Example:

BIC R3,R4

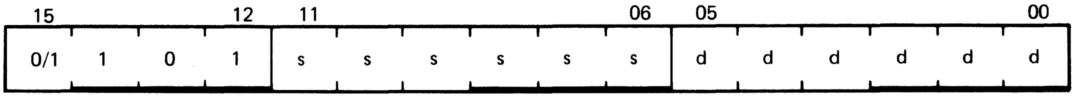
Before	After
(R3) = 001234	(R3) = 001234
(R4) = 001111	(R4) = 000101
N Z V C	N Z V C
1 1 1 1	0 0 0 1

Before: (R3) = 0 000 001 010 011 100
(R4) = 0 000 001 001 001 001

After: (R4) = 0 000 000 001 000 001

BIS
BISB

Bit Set ■5SSDD



Operation: (dst) ← (dst) $\vee$ (src)

Condition Codes:

N: set if high-order bit of result is set, cleared otherwise
Z: set if result = 0; cleared otherwise
V: cleared
C: not affected

Description:

Word: Inclusive OR operation is performed between the source and destination operands, and the result is left at the destination address (i.e., corresponding bits set in the source are set in the destination). The contents of the destination are lost.

Byte: Same

Example:

BIS R0,R1

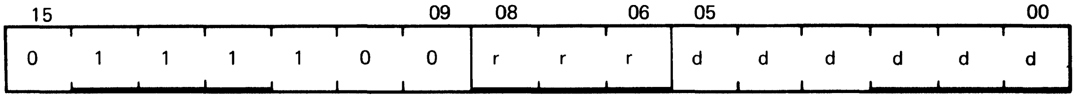
Before	After
(R0) = 001234	(R0) = 001234
(R1) = 001111	(R1) = 001335
N Z V C	N Z V C
0 0 0 0	0 0 0 0

Before: (R0) = 0 000 001 010 011 100
(R1) = 0 000 001 001 001 001

After: (R1) = 0 000 001 011 011 101

XOR

Exclusive OR **074RDD**



MR-5230

Operation: (dst) ← (dst) ∀ (Reg)

Condition Codes:

N: set if the result < 0; cleared otherwise
Z: set if result = 0; cleared otherwise
V: cleared
C: unaffected

Description: The exclusive OR of the register and destination operand is stored in the destination address. Contents of register are unaffected. Assembler format is: XOR R,D.

Example: XOR R0,R2

Before	After
(R0) = 001234	(R0) = 001234
(R2) = 001111	(R2) = 000325
N Z V C	N Z V C
1 1 1 1	0 0 0 1

Before: (R0) = 0 000 001 010 011 100
(R2) = 0 000 001 001 001 001

After: (R2) = 0 000 000 011 010 101

A.3.5 Program Control Instructions

A.3.5.1 Branches – Program control instructions cause a branch to a location defined by the sum of the offset (multiplied by two) and the current contents of the program counter if:

1. The branch instruction is unconditional.
2. The branch instruction is conditional, and the conditions are met after testing the condition codes (NZVC).

The offset is the number of words from the current contents of the PC forward or backward. The current contents of the PC point to the word following the branch instruction.

Although the offset expresses a byte address, the PC is expressed in words. Before it is added to the PC, the offset is automatically multiplied by two and sign extended to express words. Bit 7 is the sign of the offset. If it is set, the offset is negative and the branch is done in the backward direction. Similarly, if bit 7 is not set, the offset is positive and the branch is done in the forward direction.

The 8-bit offset allows branching in the backward direction by 200_8 words (400 bytes) from the current PC, and in the forward direction by 177_8 words (376 bytes) from the current PC.

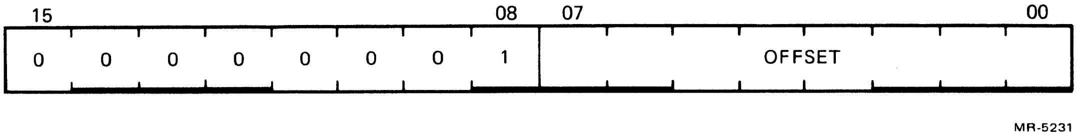
The microprocessor assembler handles address arithmetic for the user and computes and assembles the proper offset field for branch instructions in the following form.

Bxx loc

where Bxx is the branch instruction and loc is the address to which the branch is to be made. The assembler gives an error indication in the instruction if the permissible branch range is exceeded. Branch instructions have no effect on condition codes. Conditional branch instructions, where the branch condition is not met, are treated as NO OPs.

BR

Branch (Unconditional) 000400 Plus Offset



Operation: $PC \leftarrow PC + (2 \times \text{offset})$

Condition Codes: Unaffected

Description: A way of transferring program control within a range of -128_{10} to $+127_{10}$ words with a one-word instruction is provided.

New PC address = updated PC + (2 × offset)

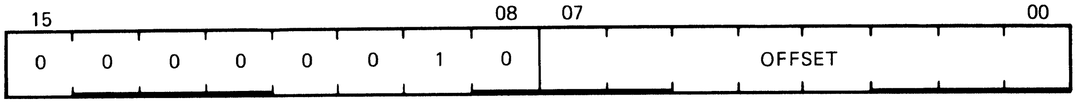
Updated PC = address of branch instruction +2

Example: With the branch instruction at location 500, the following offsets apply.

New PC Address	Offset Code	Offset (decimal)
474	375	−3
476	376	−2
500	377	−1
502	000	0
504	001	+1
506	002	+2

BNE

Branch If Not Equal (to Zero) 001000 Plus Offset



MR-5232

Operation: PC ← PC + (2 × offset) if $Z = 0$

Condition Codes: Unaffected

Description: The state of the Z-bit is tested, and a branch is caused if the Z-bit is clear. BNE is the complementary operation to BEQ. BNE is used to test inequality following a CMP, to test that some bits set in the destination were also in the source following a BIT operation, and generally, to test that the result of the previous operation was not zero.

Example:

CMP A,B	compare A and B
BNE C	branch if they are not equal

will branch to C if $A \neq B$

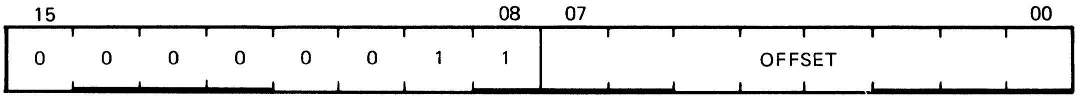
and the sequence

ADD A,B	add A to B
BNE C	branch if the result is not equal to 0

will branch to C if $A + B = 0$

BEQ

Branch If Equal (to Zero) 001400 Plus Offset



MR-5233

Operation: PC ← PC + (2 × offset) if $Z = 1$

Condition Codes: Unaffected

Description: The state of the Z-bit is tested and a branch is caused if Z is set. BEQ is used to test equality following a CMP operation, to test that no bits set in the destination were also set in the source following a BIT operation, and generally, to test that the result of the previous operation was zero.

Example:

```
CMP A,B          compare A and B
BEQ C            branch if they are equal
```

will branch to C if A = B (A − B = 0)

and the sequence

```
ADD A,B          add A to B
BEQ C            branch if the result = 0
```

will branch to C if A + B = 0

BPL

Branch If Plus

100000 Plus Offset

15							08	07							00
1	0	0	0	0	0	0	0	OFFSET							

MR-5234

Operation: PC ← PC + (2 × offset) if N = 0

Condition Codes: Unaffected

Description: The state of the N-bit is tested, and a branch is caused if N is clear (positive result). BPL is the complementary operation of BMI.

BMI

Branch If Minus

100400 Plus Offset

15							08	07							00
1	0	0	0	0	0	0	1	OFFSET							

MR-5235

Operation: PC ← PC + (2 × offset) if N = 1

Condition Codes: Unaffected

Description: The state of the N-bit is tested, and a branch is caused if N is set. BMI is used to test the sign (most significant bit) of the result of the previous operation, branching if negative. BMI is the complementary function of BPL.

BVC

Branch If Overflow Is Clear 102000 Plus Offset

15							08	07							00
1	0	0	0	0	1	0	0	OFFSET							

MR-5236

Operation: PC ← PC + (2 × offset) if V = 0

Condition Codes: Unaffected

Description: The state of the V-bit is tested, and a branch is caused if the V-bit is clear. BVC is the complementary operation to BVS.

BVS

Branch If Overflow Is Set 102400 Plus Offset

15							08	07							00
1	0	0	0	0	1	0	1	OFFSET							

MR-5237

Operation: PC ← PC + (2 × offset) if V = 1

Condition Codes: Unaffected

Description: The state of the V-bit (overflow) is tested, and a branch is caused if the V-bit is set. BVS is used to detect arithmetic overflow in the previous operation.

BCC

Branch If Carry Is Clear 103000 Plus Offset

15							08	07							00
1	0	0	0	0	1	1	0	OFFSET							

MR-5238

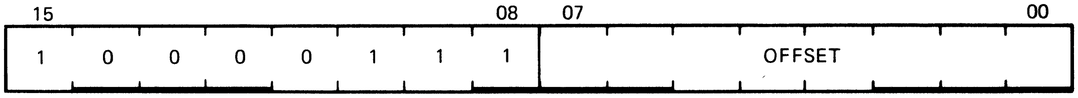
Operation: PC ← PC + (2 × offset) if C = 0

Condition Codes: Unaffected

Description: The state of the C-bit is tested, and a branch is caused if C is clear. BCC is the complementary operation to BCS.

BCS

Branch If Carry Is Set 103400 Plus Offset



MR-5239

Operation: PC ← PC + (2 × offset) if C = 1

Condition Codes: Unaffected

Description: The state of the C-bit is tested, and a branch is caused if C is set. BCS is used to test for a carry in the result of a previous operation.

A.3.5.2 Signed Conditional Branches – Particular combinations of the condition code bits are tested with the signed conditional branches. These instructions are used to test the results of instructions in which the operands were considered as signed (two's complement) values.

The sense of signed comparisons differs from unsigned comparisons. In signed 16-bit (two's complement) arithmetic, the sequence of values is as follows.

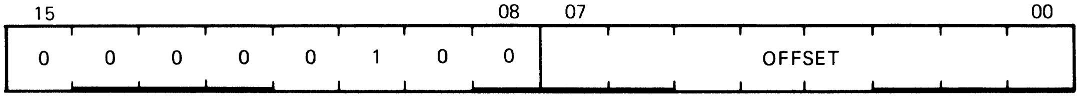
largest	077777
	077776
positive	.
	.
	.
	000001
zero	000000
	177777
	177776
	.
negative	.
	.
	100001
smallest	100000

In unsigned 16-bit arithmetic, the sequence is as follows.

highest	177777
	.
	.
	.
	.
	.
	.
	000002
	000001
lowest	000000

BGE

Branch If Greater Than or Equal (to Zero) 002000 Plus Offset



MR-5240

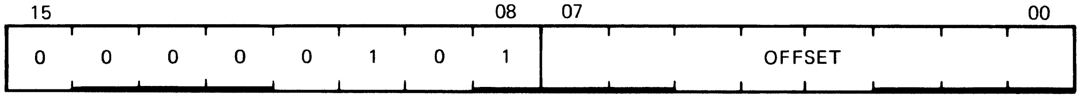
Operation: $PC \leftarrow PC + (2 \times \text{offset})$ if $N \forall V = 0$

Condition Codes: Unaffected

Description: A branch is caused if N and V are either both clear or both set. BGE is the complementary operation to BLT. Thus, BGE will always cause a branch when it follows an operation that caused addition of two positive numbers. BGE will also cause a branch on a zero result.

BLT

Branch If Less Than (Zero) 002400 Plus Offset



MR-5241

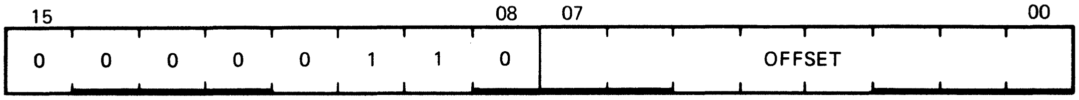
Operation: $PC \leftarrow PC + (2 \times \text{offset})$ if $N \forall V = 1$

Condition Codes: Unaffected

Description: A branch is caused if the exclusive OR of the N- and V-bits is one. Thus, BLT will always branch following an operation that added two negative numbers, even if overflow occurred. In particular, BLT will always cause a branch if it follows a CMP instruction operating on a negative source and a positive destination (even if overflow occurred). Further, BLT will never cause a branch when it follows a CMP instruction operating on a positive source and negative destination. BLT will not cause a branch if the result of the previous operation was zero (without overflow).

BGT

Branch If Greater Than (Zero) 003000 Plus Offset



MR-5242

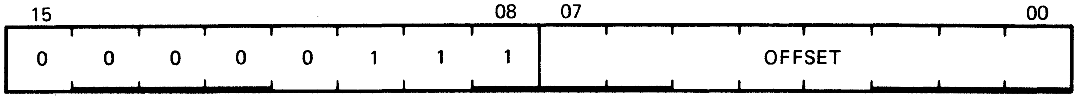
Operation: $PC \leftarrow PC + (2 \times \text{offset})$ if $Z \vee (N \forall V) = 0$

Condition Codes: Unaffected

Description: Operation of BGT is similar to BGE, however, BGT will not cause a branch on a zero result.

BLE

Branch If Less Than or Equal (to Zero) 003400 Plus Offset



MR-5243

Operation: $PC \leftarrow PC + (2 \times \text{offset})$ if $Z \vee (N \forall V) = 1$

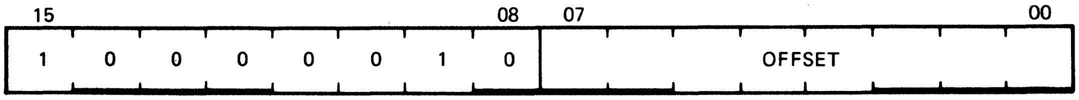
Condition Codes: Unaffected

Description: Operation is similar to BLT, however, BLE also will cause a branch if the result of the previous operation was zero.

A.3.5.3 Unsigned Conditional Branches –The unsigned conditional branches provide a means to test the results of comparison operations in which the operands are considered unsigned values.

BHI

Branch If Higher 101000 Plus Offset



MR-5244

Operation: $PC \leftarrow PC + (2 \times \text{offset})$ if $C = 0$ and $Z = 0$

Condition Codes: Unaffected

Description: A branch occurs if the previous operation did not cause a carry or a zero result. This will happen in comparison (CMP) operations as long as the source has a higher unsigned value than the destination.

BLOS

Branch If Lower or Same 101400 Plus Offset

15							08	07							00
1	0	0	0	0	0	1	1	OFFSET							

MR-5245

Operation: PC ← PC + (2 × offset) if C V Z = 1

Condition Codes: Unaffected

Description: A branch occurs if the previous operation caused either a carry or a zero result. BLOS is the complementary operation to BHI. The branch will occur in comparison operations as long as the source is equal to, or has a lower unsigned value than the destination.

BHIS

Branch If Higher or Same 103000 Plus Offset

15							08	07							00
1	0	0	0	0	1	1	0	OFFSET							

MR-5246

Operation: PC ← PC + (2 × offset) if C = 0

Condition Codes: Unaffected

Description: BHIS is the same instruction as BCC. This mnemonic is included for convenience only.

BLO

Branch If Lower 103400 Plus Offset

15							08	07							00
1	0	0	0	0	1	1	1	OFFSET							

MR-5247

Operation: PC ← PC + (2 × offset) if C = 1

Condition Codes: Unaffected

Description: BLO is the same instruction as BCS. This mnemonic is included for convenience only.

A.3.5.4 Jump and Subroutine Instructions – The subroutine call in the microprocessor provides for automatic nesting of subroutines, re-entrance, and multiple entry points. Subroutines may call other subroutines (or themselves) to any level of nesting without making special provisions for storage of return addresses at each level of subroutine call. The subroutine calling mechanism does not modify any fixed location in memory, and thus, provides for re-entrance. This allows one copy of a subroutine to be shared among several interrupting processes.

JMP

Jump

0001DD

15									06	05					00
0	0	0	0	0	0	0	0	0	1	d	d	d	d	d	d

MR-5248

Operation: PC ← (dst)

Condition Codes: Unaffected

Description: More flexible program branching than that available with the branch instructions is provided. Control may be transferred to any location in memory (no range limitation) and can be accomplished with the full flexibility of the addressing modes, with the exception of register mode 0. Execution of a jump with mode 0 will cause an illegal instruction condition, and will cause the CPU to trap to vector address 4. (Program control cannot be transferred to a register.) Register deferred mode is legal and will cause program control to be transferred to the address held in the specified register. Instructions are word data and therefore, must be fetched from an even-numbered address.

Deferred index mode JMP instructions permit transfer of control to the address contained in a selectable element of a table of dispatch vectors.

Example:

	JMP FIRST	transfers to FIRST
First:		
	JMP @LIST	transfers to location pointed to at LIST
List:	FIRST	pointer to FIRST
	JMP @(SP)+	transfers to location pointed to by the top of the stack and removes the pointer from the stack

JSR

Jump to Subroutine 004RDD

15						09	08		06	05					00
0	0	0	0	1	0	0	r	r	r	d	d	d	d	d	d

MR-5249

Operation: (tmp) ← (dst) (tmp is an internal processor register)

↓ (SP) ← reg (push reg contents onto processor stack)

reg ← PC (PC holds location following JSR; this address is now put in reg)

PC ← (dst) (PC now points to subroutine destination)

Condition Codes: Unaffected

Description: The old contents of the specified register (the linkage pointer) are automatically pushed onto the processor stack, and new linkage information is placed in the register. Thus, subroutines nested within subroutines to any depth may all be called with the same linkage register. There is no need either to plan the maximum depth at which any particular subroutine will be called or to include instructions in each routine to save and restore the linkage pointer. Further, since all linkages are saved in a re-entrant manner on the processor stack, execution of a subroutine may be interrupted, and the same subroutine re-entered and executed by an interrupt service routine. Execution of the initial subroutine can then be resumed when other requests are satisfied. This process (called nesting) can proceed to any level.

A subroutine called with a JSR reg,(dst) instruction can access the arguments following the call with either autoincrement addressing, (reg)+, (if arguments are accessed sequentially) or by indexed addressing, X(reg), (if arguments are accessed in random order). These addressing modes may also be deferred, @(reg)+ and @X(reg), if the parameters are operand addresses rather than the operands themselves.

JSR PC,(dst) is a special case of the microprocessor subroutine call and is used for subroutine calls that transmit parameters through the general-purpose registers. The SP and the PC are the only registers that may be modified by this call.

JSR PC,@(SP)+ is another special case of the JSR instruction. It exchanges the top element of the processor stack and the contents of the program counter. This instruction is used to allow two routines to swap program control and resume operation when recalled where they left off. Such routines are called co-routines.

Return from a subroutine is done with the RTS instruction. RTS reg loads the contents of reg into the PC and pops the top element of the processor stack into the specified register.

Example:

```
                                                     R5              R6    R7
SBCALL        : JSR R5, SBR                          #1              n     SBCALL
SBCALL+4      : ARG 1
                ARG 2
                 .
                 .
SBCALL+2+2M   : ARG M
CONT          : NEXT INSTRUCTION                     #1              n     CONT
                 .
                 .
 SBR          : MOV(R5)+,dst1                        SBCALL+4        n-2   SBR
                MOV(R5)+,dst2
                 .
                MOV(R5)+,dst2
                 .
                MOV(R5)+,dstM                        SBCALL+2+2M
                OTHER INSTRUCTIONS                   CONT
EXIT          : RTS R5                               CONT            n-2   EXIT
```

This example is illustrated in the following figure.

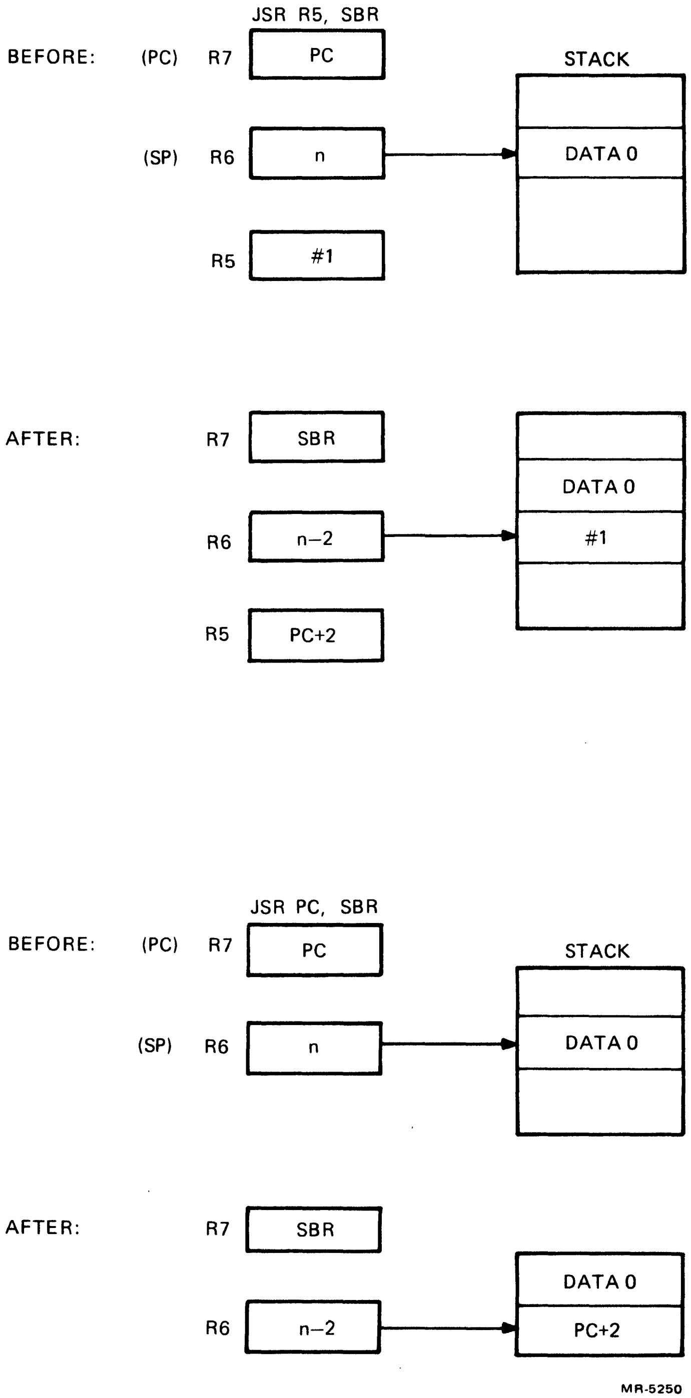


Figure A-34 JSR Example

RTS

Return from Subroutine

00020R

15												03	02		00
0	0	0	0	0	0	0	0	1	0	0	0	0	r	r	r

MR-5251

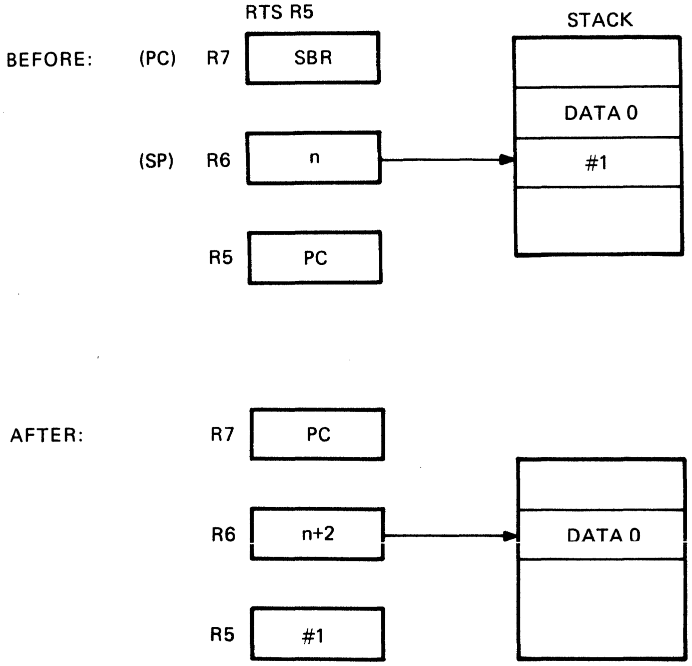
Operation: PC ← (reg)
(reg) ← (SP) ↑

Condition Codes: Unaffected

Description: Contents of register are loaded into PC, and the top element of the processor stack is popped into the specified register. Return from a nonre-entrant subroutine is typically made through the same register that was used in its call. Thus, a subroutine called with a JSR PC,(dst) exits with an RTS PC. A subroutine called with a JSR R5,(dst) may pick up parameters with addressing modes (R5)+, X(R5), or @X(R5) and finally exit with an RTS R5.

Example:

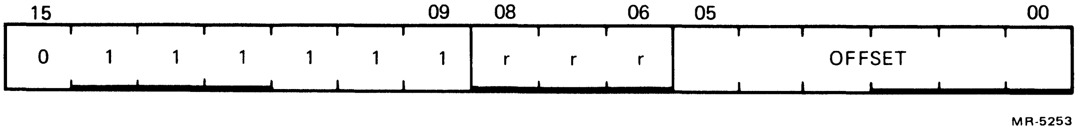
RTS R5



MR-5252

SOB

Subtract One and Branch (If $\neq 0$) 077RNN



Operation: $(R) \leftarrow (R) - 1$; if this result $\neq 0$ then $PC \leftarrow PC - (2 \times \text{offset})$; if $(R) = 0$ then $PC \leftarrow PC$

Condition Codes: Unaffected

Description: The register is decremented. If it is not equal to zero, twice the offset is subtracted from the PC (now pointing to the following word). The offset is interpreted as a 6-bit positive number. SOB provides a fast, efficient method of loop control. The assembler syntax is:

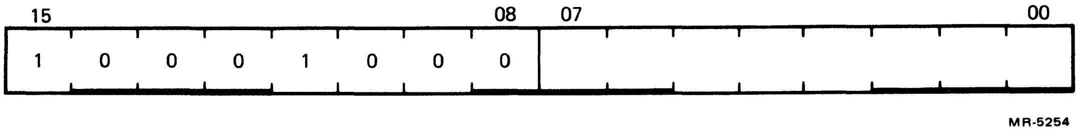
SOB R,A

where A is the address to which transfer is to be made if the decremented R is not equal to zero. The SOB instruction cannot be used to transfer control in the forward direction.

A.3.5.5 **Traps** – Trap instructions provide for calls to emulators, I/O monitors, debugging packages, and user-defined interpreters. A trap is effectively an interrupt generated by software. When a trap occurs the contents of the current program counter (PC) and processor status (PS) are pushed onto the processor stack and replaced by the contents of a two-word trap vector containing a new PC and PS. The return sequence from a trap involves executing an RTI or RTT instruction that restores the old PC and PS by popping them from the stack. Trap instruction vectors are located at permanently assigned fixed addresses.

EMT

Emulator Trap 104000–104377



Operation:

↓ (SP) ← PS
↓ (SP) ← PC
PC ← (30)
PS ← (32)

Condition Codes:

N: loaded from trap vector
Z: loaded from trap vector
V: loaded from trap vector
C: loaded from trap vector

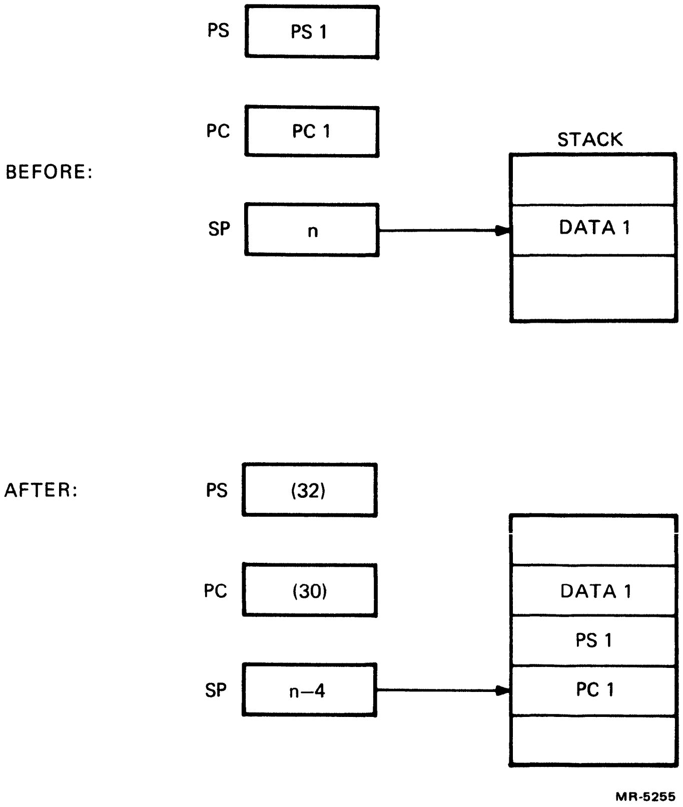
Description:

All operation codes from 104000 to 104377 are EMT instructions and may be used to transmit information to the emulating routine (e.g., function to be performed). The trap vector for EMT is at address 30. The new PC is taken from the word at address 30, and the new processor status (PS) is taken from the word at address 32.

CAUTION

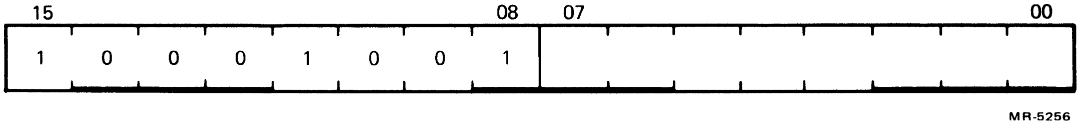
EMT is used frequently by Digital system software and is not recommended for general use.

Example:



TRAP

Trap 104400–104777



Operation:

↓ (SP) ← PS
↓ (SP) ← PC
PC ← (34)
PS ← (36)

Condition Codes:

N: loaded from trap vector
Z: loaded from trap vector
V: loaded from trap vector
C: loaded from trap vector

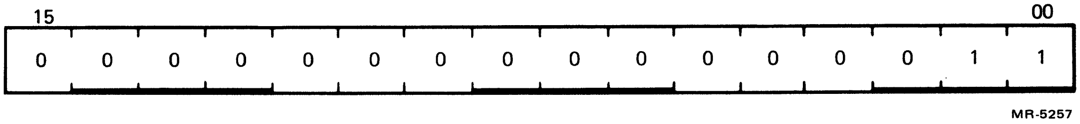
Description:

Operation codes from 104400 to 104777 are TRAP instructions. TRAP and EMT instructions are identical in operation, however, the trap vector for TRAP is at address 34.

NOTE
Because Digital software makes frequent use of EMT, the TRAP instruction is recommended for general use.

BPT

Breakpoint Trap 000003



Operation:

↓ (SP) ← PS
↓ (SP) ← PC
PC ← (14)
PS ← (16)

Condition Codes:

N: loaded from trap vector
Z: loaded from trap vector
V: loaded from trap vector
C: loaded from trap vector

Description:

A trap sequence with a trap vector address of 14 is performed. BPT is used to call debugging aids. The user is cautioned against employing code 000003 in programs run under these debugging aids.

(No information is transmitted in the low byte.)

IOT

Input/Output Trap 000004

15															00
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

MR-5258

Operation: ↓ (SP) ← PS
↓ (SP) ← PC
PC ← (20)
PS ← (22)

Condition Codes: N: loaded from trap vector
Z: loaded from trap vector
V: loaded from trap vector
C: loaded from trap vector

Description: A trap sequence with a trap vector address of 20 is performed.

(No information is transmitted in the low byte.)

RTI

Return from Interrupt 000002

15															00
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

MR-5260

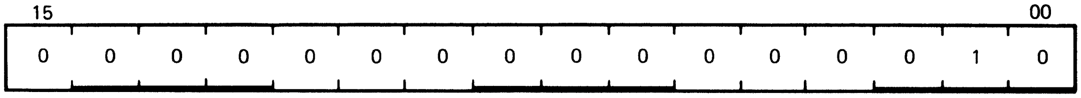
Operation: PC ← (SP) ↑
PS ← (SP) ↑

Condition Codes: N: loaded from processor stack
Z: loaded from processor stack
V: loaded from processor stack
C: loaded from processor stack

Description: Used to exit from an interrupt or TRAP service routine. The PC and PS are restored (popped) from the processor stack. If a trace trap is pending, the first instruction after RTI will not be executed prior to the next T trap.

RTT

Return from Trap 000006



MR-5259

Operation: PC ← (SP) ↑
PS ← (SP) ↑

Condition Codes:
N: loaded from processor stack
Z: loaded from processor stack
V: loaded from processor stack
C: loaded from processor stack

Description: Operation is the same as RTI, however, RTT inhibits a trace trap while RTI permits a trace trap. If new PS has T-bit set, trap will occur after execution of first instruction after RTT.

A.3.5.6 Reserved Instruction Traps – Reserved instruction traps are caused by attempts to execute instruction codes reserved for future processor expansion (reserved instructions) or instructions with illegal addressing modes (illegal instructions). Order codes not corresponding to any of the instructions described are reserved instructions. JMP and JSR with register mode destinations are illegal instructions and trap to vector address 4. Reserved instructions trap to vector address 10.

A.3.5.7 HALT Interrupt – The HALT interrupt is caused by the –HALT line. The –HALT interrupt saves the PC and PS and goes to the restart address with PS = 340.

A.3.5.8 Trace Trap –The trace trap is enabled by bit 4 of the PS and causes processor traps at the end of instruction execution. The instruction that is executed after the instruction that set the T-bit will proceed to completion and then trap through the trap vector at address 14. The trace trap is a system debugging aid and is transparent to the general programmer.

A.3.5.9 Power Failure Interrupt –The power failure interrupt occurs when –PF line is asserted. Vectors for power failure are locations 24 and 26. Trap will occur if an RTI instruction is executed in a power fail service routine.

A.3.5.10 Interrupts – See Table 3-1.

NOTE
Bit 4 of the processor status can only be set indirectly by executing an RTI or RTT instruction with the desired PS on the stack.

A.3.5.11 Special Cases (T-bit) – The following are special cases of the T-bit.

NOTE

The traced instruction follows the instruction that sets the T-bit.

1. An instruction that cleared the T-bit – Upon fetching the traced instruction, an internal flag, the trace flag, was set. The trap will still occur at the end of execution of this instruction. The status word on the stack, however, will have a clear T-bit.

2. An instruction that set the T-bit – Because the T-bit was already set, setting it again has no effect. The trap will occur.

3. An instruction that caused an instruction trap – The instruction trap is performed, and the entire routine for the service trap is executed. If the service routine exits with an RTI or in any other way restores the stacked status word, the T-bit is set again, the instruction following the traced instruction is executed, and, unless it is one of the special cases noted previously, a trace trap occurs.

4. Interrupt trap priorities – When multiple trap and interrupt conditions occur simultaneously, the following order of priorities is observed (from high to low).

    1. Halt line
    2. Power fail trap
    3. Trace trap
    4. Internal interrupt request
    5. External interrupt request
    6. Instruction traps

A.3.6 Miscellaneous Instructions

HALT

Halt 000000

15															00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MR-5261

Operation:

↓ (SP) ← PS
↓ (SP) ← PC
PC ← restart address
PS ← 340

Condition Codes: Unaffected

Description: The processor goes to the restart address after placing the current PC and PS on the stack. PS is initialized to 340.

WAIT

Wait for Interrupt 000001

15															00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

MR-5262

Condition Codes: Unaffected

Description: In WAIT, as in all instructions, the PC points to the next instruction following the WAIT instruction. Thus, when an interrupt causes the PC and PS to be pushed onto the processor stack, the address of the next instruction following the WAIT is saved. The exit from the interrupt routine (i.e., execution of an RTI instruction) will cause resumption of the interrupted process at the instruction following the WAIT.

RESET

Reset External Bus 000005

15															00
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

MR-5263

Condition Codes: Unaffected

Description: The –BCLR line is asserted and the mode register is loaded. –BCLR is negated, and an ASPI transaction takes place. PC, PS, and R0–R5 are not affected.

MFPT

Move from Processor Type Word 000007

15															00
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

MR-7198

Operation: R0 ← 4

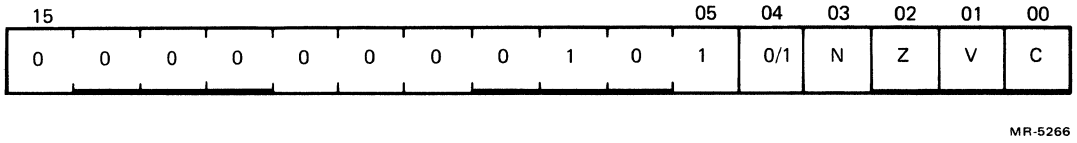
Condition Codes: Unaffected

Description: The number four is placed in R0 telling the system software that the processor type is Micro/T-11.

A.3.7 Condition Code Operators

CLN SEN
CLZ SEZ
CLV SEV
CLC SEC
CCC SCC

Condition Code Operators 0002XX



Description:

Condition code bits are set and cleared. Selectable combinations of these bits may be cleared or set together. Condition code bits corresponding to bits in the condition code operator (bits 0–3) are modified according to the sense of bit 4, the set/clear bit of the operator (i.e., set the bit specified by bit 0, 1, 2, or 3, if bit 4 is a one). Corresponding bits are cleared if bit 4 = 0.

Mnemonic	Operation	OP Code
CLC	Clear C	000241
CLV	Clear V	000242
CLZ	Clear Z	000244
CLN	Clear N	000250
SEC	Set C	000261
SEV	Set V	000262
SEZ	Set Z	000264
SEN	Set N	000270
SCC	Set all CCs	000277
CCC	Clear all CCs	000257
	Clear V and C*	000243
NOP	No operation	000240

*Combinations of the above set or clear operations may be ORed together to form combined instructions. Clear V and C represents CLC (241) ORed with CLV (code 242).

APPENDIX B
PROGRAMMING DIFFERENCES

Table B-1 presents a concise comparison of the KXT11-CA SBC, LSI-11/2, and LSI-11/23 modules.

Table B-1 KXT11-CA, LSI-11/2, and LSI-11/23 Comparisons

Activity	KXT11-CA	LSI-11/2	LSI-11/23
OPR %R,(R)+ or OPR %R,–(R) using the same register as both source and destination: contents of 'R' are incremented (decremented) by two before being used as the source operand.	X		X
OPR %R,@(R)+ or OPR %R,@–(R) using the same register as both source and destination: contents of 'R' are incremented (decremented) by two before being used as the source operand.	X		X
In the previous two cases, initial contents of 'R' are used as the source operand.		X	
OPR PC,X(R); OPR PC,@X(R); OPR PC,@A; or OPR PC,A: location A will contain the PC of OPR + 4.	X		X
In the previous case, location A will contain the PC of OPR + 2.		X	
JMP (R)+ or JSR reg,(R)+: initial contents of 'R' are used as the new PC.	X	X	X
JMP %R or JSR reg,%R traps to 4 (illegal instruction).	X	X	X
Only one LSI-11 bus interrupt level (BR4) exists.	X	X	

Table B-1 KXT11-CA, LSI-11/2, and LSI-11/23 Comparisons (Cont)

Activity	KXT11-CA	LSI-11/2	LSI-11/23
Four local interrupt levels exist.	X		
Four LSI-11 interrupt levels exist.			X
Stack overflow not implemented.	X	X	
A stack overflow trap exists.			X
The first instruction in an interrupt routine will not be executed if another interrupt occurs at a higher priority level than assumed by the first interrupt.	X	X	X
Eight general-purpose registers.	X	X	X
PSW address 177776 not implemented. Must use MTPS and MFPS instructions.	X	X	
Only implicit references (RTI, RTT, traps, and interrupts) can load T-bit. Console cannot load T-bit.	X	X	X
If an interrupt occurs during an instruction that has the T-bit set, the T-bit trap is acknowledged before the interrupt.	X	X	X
If RTI sets the T-bit, T-bit trap is acknowledged immediately following RTI.	X	X	X
T-bit trap will sequence out of WAIT instruction.	X		X
If RTT sets the T-bit, the T-bit trap occurs after the instruction following RTT.	X	X	X
RESET instruction consists of 17 μs of INIT followed by a minimum 3.2 μs pause. Power fail is not recognized until the instruction is complete.	X		
Odd address references using the SP do not trap.	X		

Table B-1 KXT11-CA, LSI-11/2, and LSI-11/23 Comparisons (Cont)

Activity	KXT11-CA	LSI-11/2	LSI-11/23
Nonexistent address references using the SP trap to the restart address.	X		
MOVB instruction does a read (DATI) and a write (DATO) bus sequence for last memory cycle.		X	
MOV instruction does a write (DATO) bus sequence for the last memory cycle.		X	X
MOV instruction does a read (DATI) and a write (DATO) bus sequence for last memory cycle.	X		
CLR(B) and SXT do a read (DATI) and a write (DATO) sequence for the last bus cycle.	X		
CLR(B) and SXT do a read (DATI) and a write (DATO) bus sequence for the last bus cycle.		X	
CLR(B) and SXT do a write (DATO) bus sequence for the last bus cycle.			X
MARK instruction.		X	X
SOB, RTT, SXT, XOR instructions.	X	X	X
SWAB clears V.	X	X	X
ASH, ASHC, DIV, MUL instructions.		X	X
Register addresses (177700–177717) are handled as regular memory addresses. No internal registers are addressable from either the bus or the console.	X		
Register addresses (177000–177717) timeout when used as program addresses by the CPU.		X	X
If PC contains a nonexistent memory address and a bus error occurs, PC will have been incremented.	X	X	X

Table B-1 KXT11-CA, LSI-11/2, and LSI-11/23 Comparisons (Cont)

Activity	KXT11-CA	LSI-11/2	LSI-11/23
If register contains a nonexistent memory address in mode 2 and a bus error occurs, register will be incremented.	X	X	X
If register contains an odd value in mode 2 and a bus error occurs, register will be incremented.	X	X	X
HALT in user mode traps to 10.			X
HALT instruction pushes PC and PSW on the stack and loads the PSW with 340 and the PC with the restart address.	X		
Only power-up mode 2 implemented.	X		
Resident ODT microcode.		X	X
Instruction execution runs to completion regardless of bus error.	X		
BEVNT line interrupt on level 6.	X		X
Bus error traps to restart address. Instruction runs to completion before trap.	X		
Bus error during IAK vectors through 0 and traps to restart address. The first instruction of service routine is guaranteed to execute.	X		
Only 16-bit addressing supported.	X	X	
The no-BSACK 18 μs timeout implemented. If timeout occurs BDMGO aborted.			X
Bus halt line is a jumper configured nonmaskable interrupt. Acknowledgement causes PC and PSW to be stacked and the processor vectors through level 7 internal vector 140.	X		

Table B-1 KXT11-CA, LSI-11/2, and LSI-11/23 Comparisons (Cont)

Activity	KXT11-CA	LSI-11/2	LSI-11/23
Vector address accepted only on BDAL<7:2>. This limits vector address space to 374.	X		
Certain vector addresses are reserved for local devices other than BEVNT.	X		

T-11 Internal Priorities

Priority of DMA, system traps, external interrupts, internal interrupts, HALT trap, and WAIT are as follows.

DMA	(highest priority)
HALT trap (timeout request)	
Power-fail trap	
Traps (illegal instruction, T-bit, EMT)	
Internal interrupt request	
External interrupt request	
WAIT instruction	(lowest priority)

KXT11-C Priorities

Priority of system traps, external interrupts, internal interrupts, HALT trap, and WAIT are as follows.

HALT trap (timeout request)	(highest priority)
Power-Fail trap	
Traps (illegal instruction, T-bit, EMT)	
Internal vector	
External vector	
WAIT instruction	
TEST	(lowest priority)

APPENDIX C
PERFORMANCE PARAMETERS

The fetch and execute times listed in Table C-1 assume that the KXT11-CA is transacting with local devices that do not require cycle slips when accessed.

The measure of Q-Bus interrupt latency is the time from the assertion of BIRQ until BIAKI is accepted by the interrupting device electrically closest to the processor on the Q-Bus.

The measure of local interrupt latency is the time from assertion of the request until the time the microprocessor is ready to fetch the first instruction in the interrupt service routine. This time is primarily comprised of the time to perform two pushes and a PC and PSW restore.

Interrupt Latency:	LOCAL	23.2 μs
	Q-BUS	9.3 μs

NOTE

Assume that the stack and vector memory reside on the KXT11-CA and that the Q-Bus device can assert BRPLY and vector within 600 ns after receiving IAKI. The service latency (time from BIRQ until the time the microprocessor is ready to fetch the first instruction in the interrupt service routine) depends on the response time of the interrupting device (i.e., RDIN to TRPLY and negation of TRPLY).

DMA latency is the period of time between a device asserting its BDMR and receiving BDMGI when it resides on the Q-Bus as the electrically closest DMA device to the processor.

DMA latency:	1.3 μs (minimum)	11.0 μs (maximum)

WAIT instruction latencies:

Internal vector:	11.8 μs
External vector:	12.4 μs
DMA:	5.06 μs

Table C-1 Instruction Timing

Single Operand Instructions	Destination Mode	Fetch and Execute Time (μs)	Number of Bus Transactions	Number of Microcycles
CLR(B), COM(B),	0	2.44	1	4
INC(B), DEC(B),	1	4.27	3	7
NEG(B), ROR(B),	2	4.27	3	7
ROL(B), ASR(B),	3	5.49	4	9
ASL(B), SWAP,	4	4.88	3	8
ADC(B), SBC(B),	5	6.10	4	10
SXT, MFPS,	6	6.10	4	10
XOR	7	7.32	5	12
	0	2.44	1	4
	1	3.66	2	6
	2	3.66	2	6
TST(B)	3	5.49	3	8
	4	4.27	2	7
	5	5.49	3	9
	6	5.49	3	9
	7	6.71	4	11
	0	4.88	1	8
	1	6.10	2	10
	2	6.10	2	10
MTPS	3	7.32	3	12
	4	6.71	2	11
	5	7.93	3	13
	6	7.93	3	13
	7	9.16	4	15

Double Operand Instructions	Source Mode	Source Mode Time (μs) Includes Fetch	Number of Bus Transactions	Number of Microcycles
MOV(B), CMP(B),	0	1.83	1	3
ADD, SUB,	1	3.05	2	5
BIT(B), BIC(B),	2	3.05	2	5
BIS(B)	3	4.27	3	7
	4	3.66	2	6
	5	4.88	3	8
	6	4.88	3	8
	7	6.10	4	10

Table C-1 Instruction Timing (Cont)

Double Operand Instructions	Destination Mode	Destination Mode Time (μs)	Number of Bus Transactions	Number of Microcycles
MOV(B), CMP(B), ADD, SUB, BIT(B), BIC(B), BIS(B)	0	0.61	0	1
	1	2.44	2	4
	2	2.44	2	4
	3	3.66	3	6
	4	3.05	2	5
	5	4.27	3	7
	6	4.27	3	7
	7	5.49	4	9
CMP(B), BIT(B)	0	0.61	0	1
	1	1.83	1	3
	2	1.83	1	3
	3	3.05	2	5
	4	2.44	1	4
	5	3.66	2	6
	6	3.66	2	6
	7	4.88	3	8

Jump and Subroutine Instructions	Destination Mode	Fetch and Execute Time (μs)	Number of Bus Transactions	Number of Microcycles
JMP	1	3.05	2	5
	2	3.66	2	6
	3	3.66	3	6
	4	3.66	2	6
	5	4.27	3	7
	6	4.27	3	7
	7	5.49	4	9
JSR	1	5.49	4	9
	2	6.10	4	10
	3	6.10	5	10
	4	6.10	4	10
	5	6.71	5	11
	6	6.71	5	11
	7	7.90	6	13
RTS	NA	4.27	2	7
SOB	NA	3.66	1	6

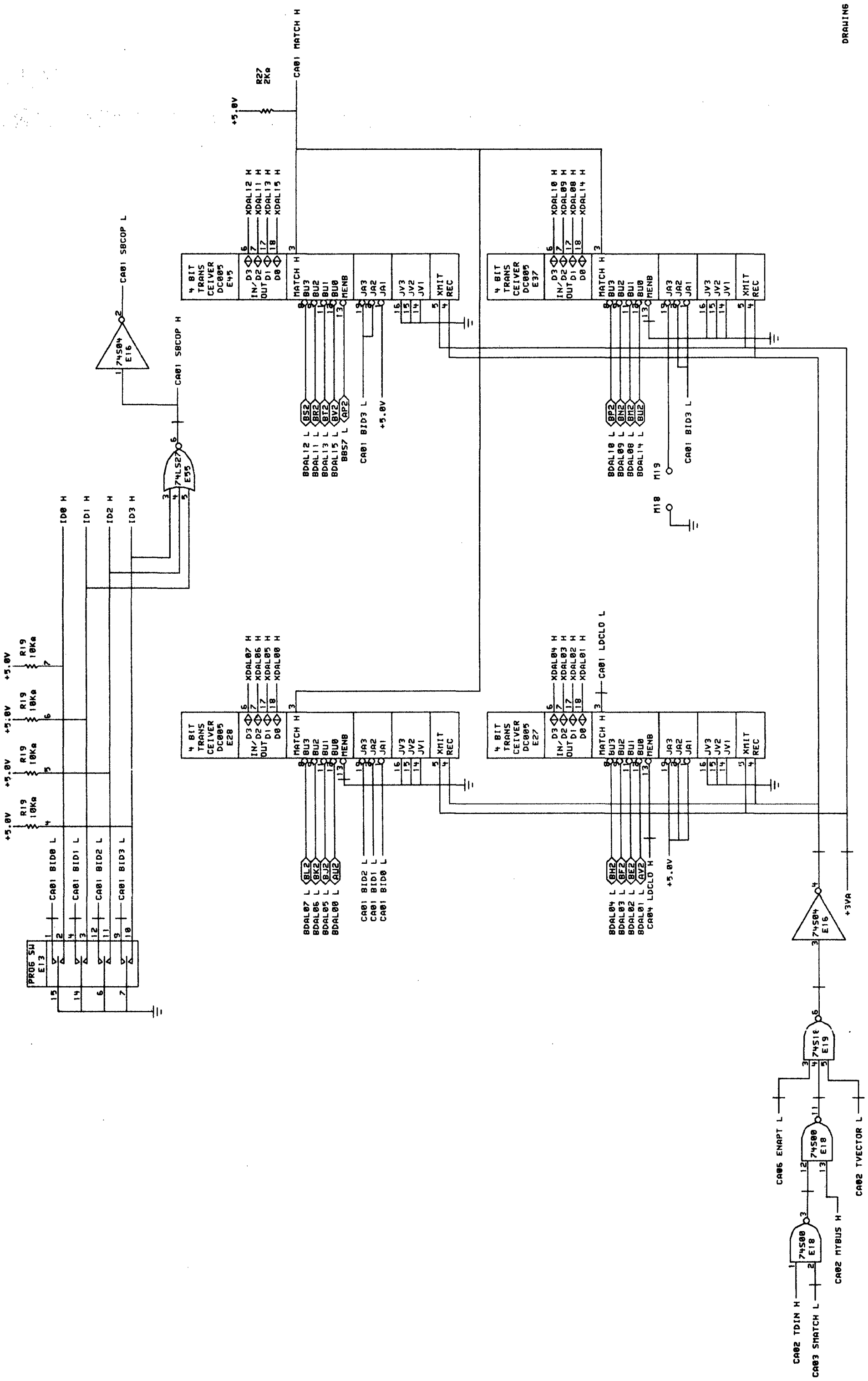
Table C-1 Instruction Timing (Cont)

Branch, Trap, and Interrupt Instructions	Destination Mode	Fetch and Execute Time (μs)	Number of Bus Transactions	Number of Microcycles
BR, BNE, BEQ, BPL, BMI, BVC, BVS, BCC, BCS, BGE, BLT, BGT, BLE, BHI, BLOS, BHIS, BLO	NA	2.44	1	4
EMT, TRAP, BPT, IOT	NA	9.77	7	16
RTI	NA	4.88	3	8
RTT	NA	6.71	3	11

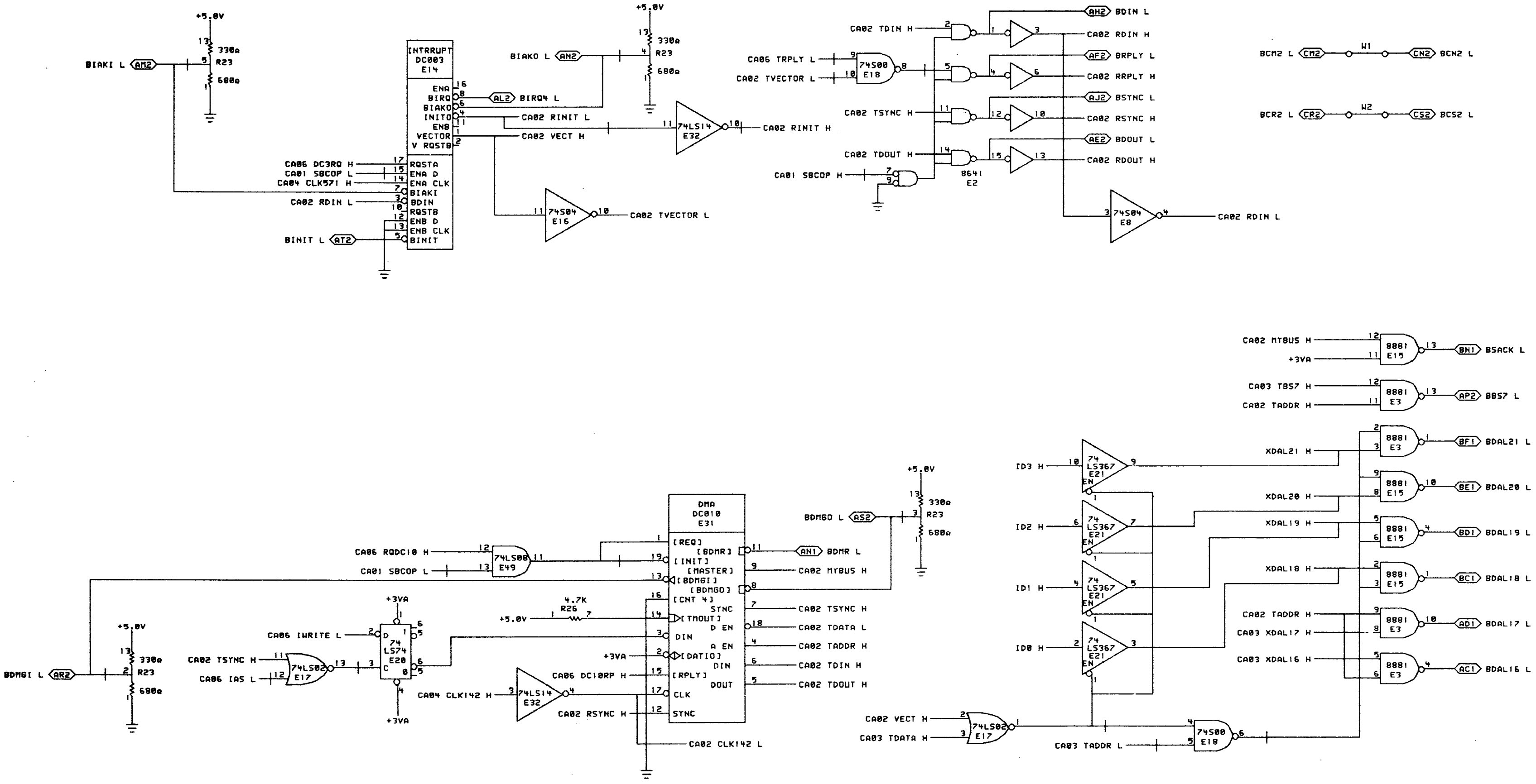
Miscellaneous and Condition Code Instructions	Destination Mode	Fetch and Execute Time (μs)	Number of Bus Transactions	Number of Microcycles
HALT	NA	8.54	5	14
WAIT	NA	2.44	1	4 then loop
RESET	NA	22.28	1	39
NOP	NA	3.66	1	6
CLC, CLV, CLZ, CLN, CCC, SEC, SEV, SEZ, SEN, SCC	NA	3.66	1	6
MFPT	NA	3.05	1	5

APPENDIX D
KXT11-CA SBC SCHEMATIC DRAWINGS

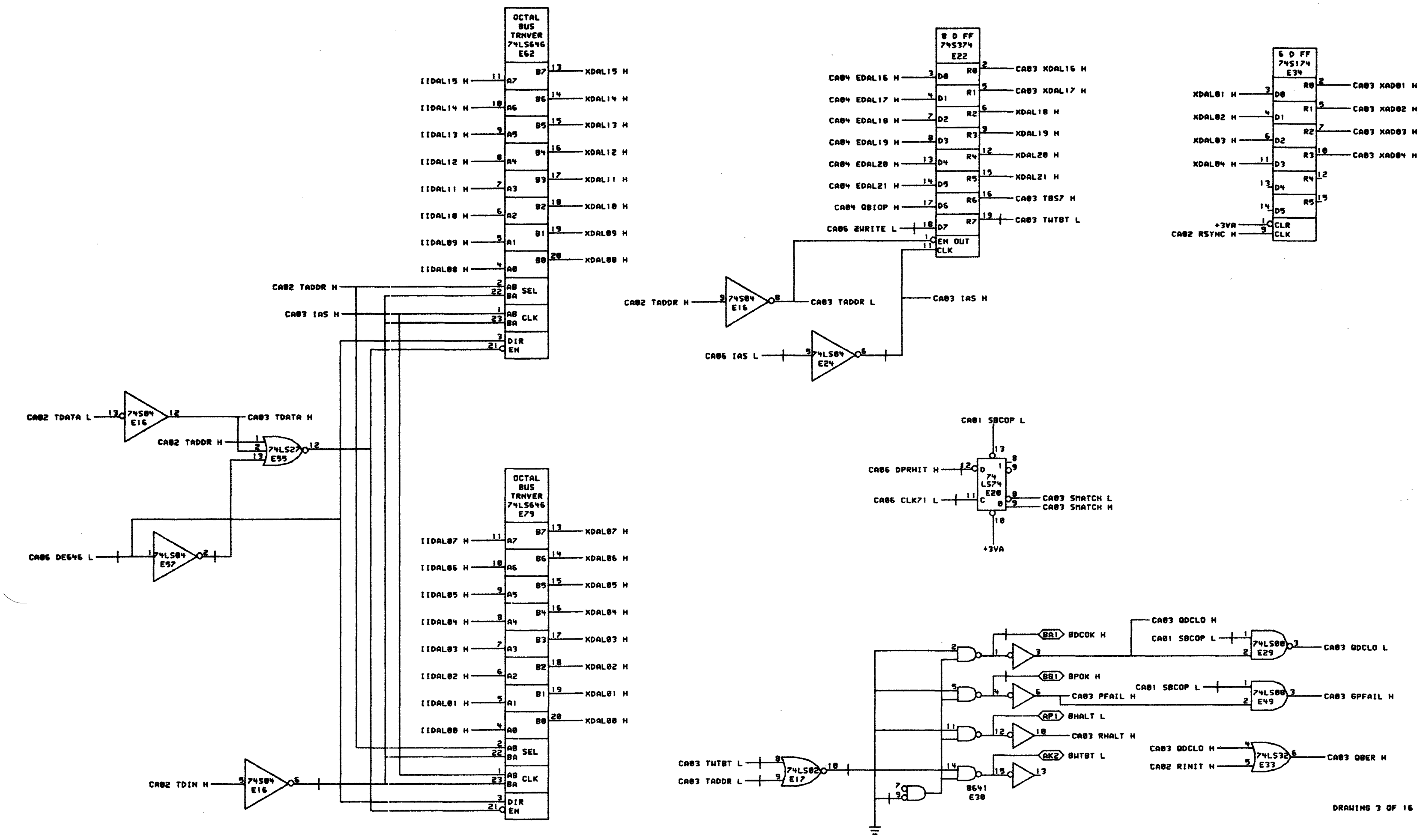
The following is the complete KXT11-CA SBC schematic drawing set.



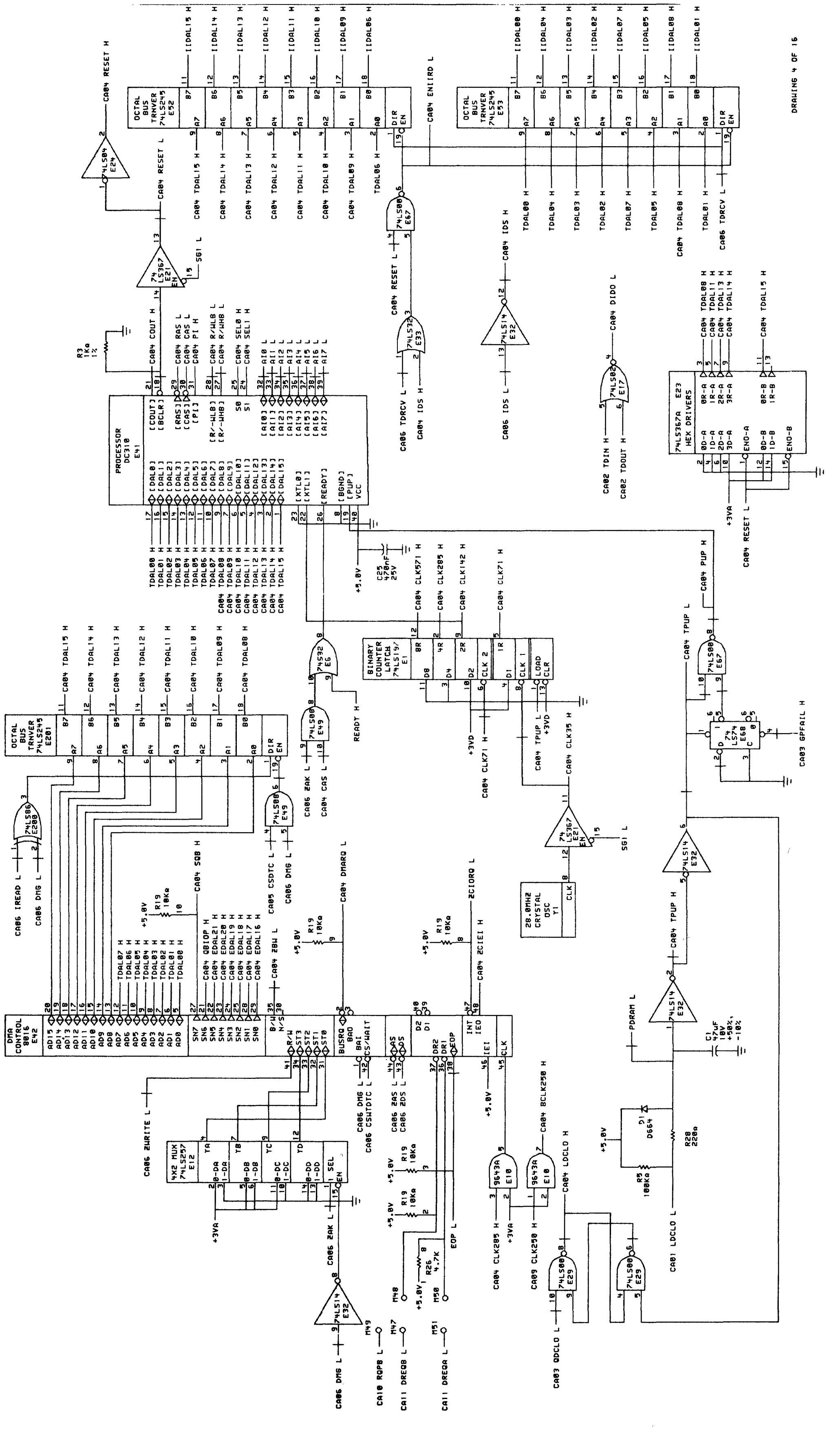
DRAWING 1 OF 16

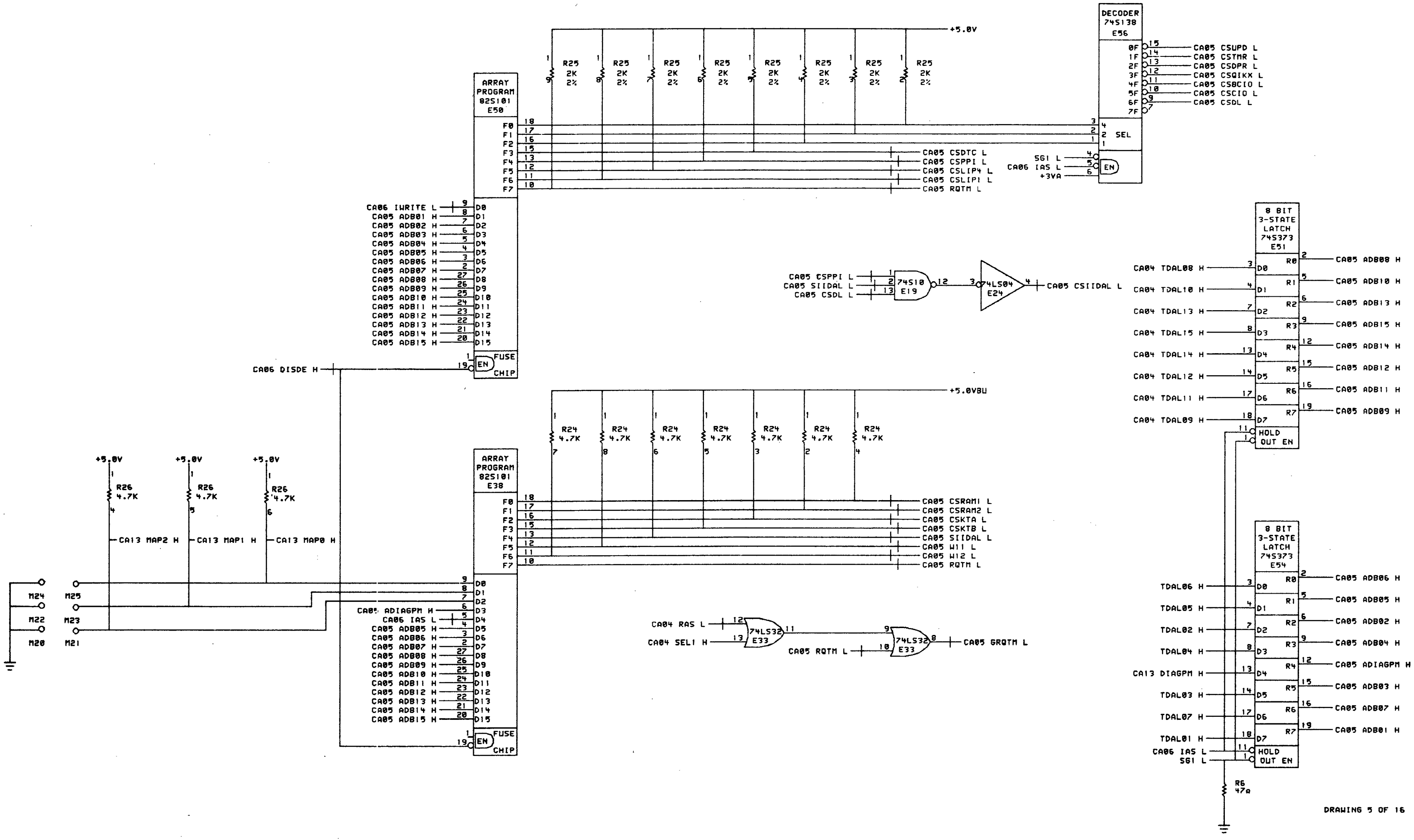


DRAWING 2 OF 16

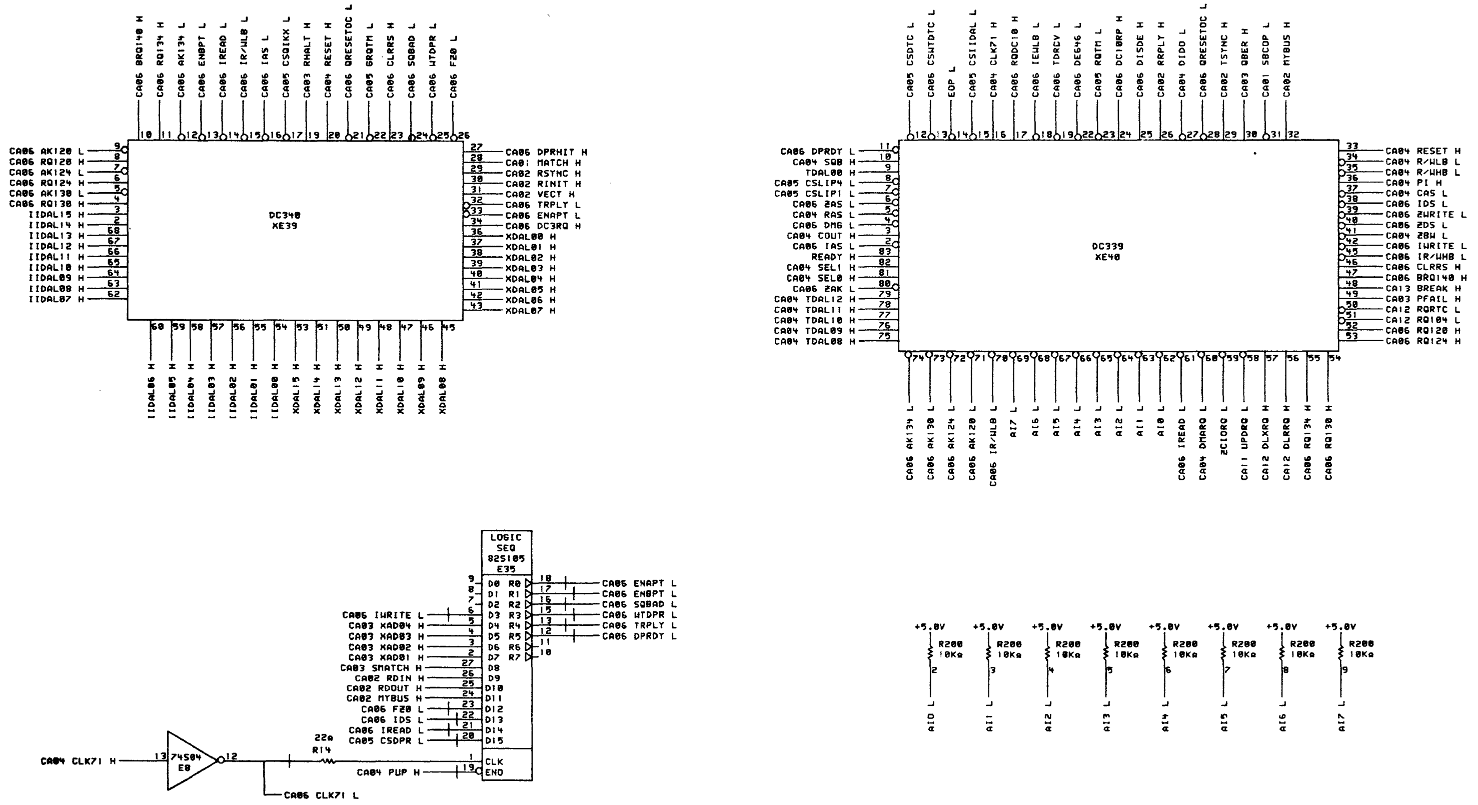


DRAWING 3 OF 16

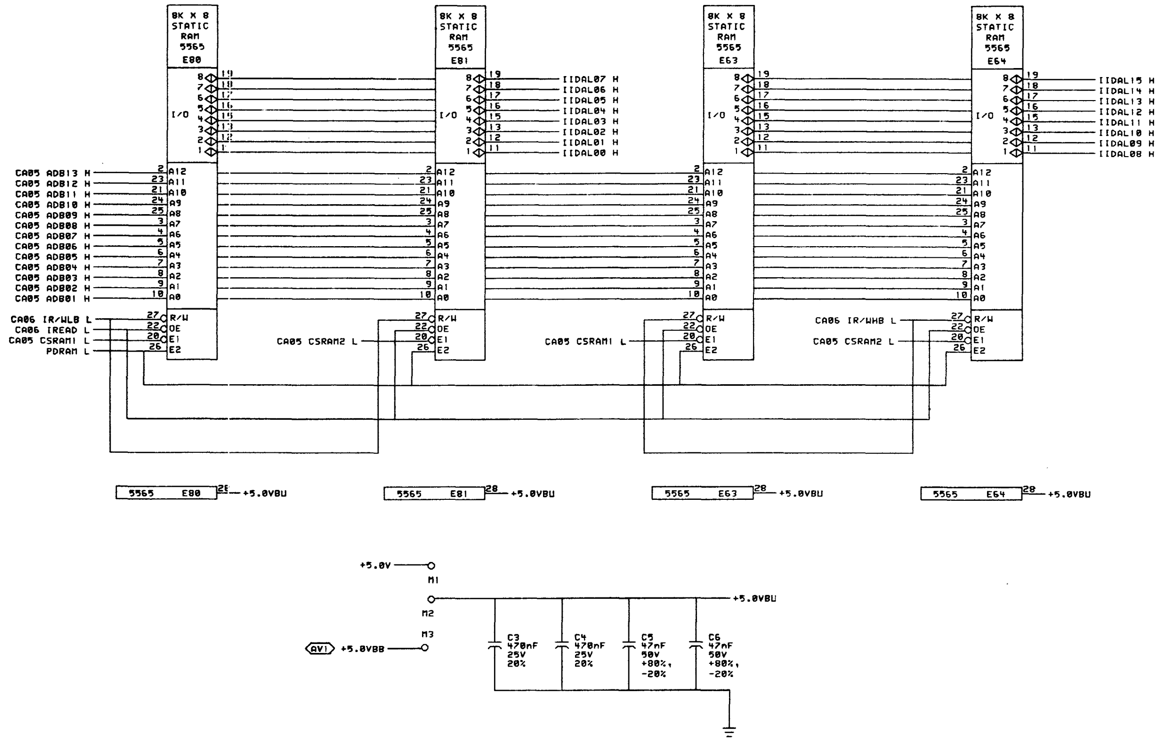

DRAWING 4 OF 16
PROCESSOR DC310 E41
DMA CONTROL 8016 E42
OCTAL BUS TRNVER 74LS245 E52
OCTAL BUS TRNVER 74LS245 E53
OCTAL BUS TRNVER 74LS245 E201
74LS367A E23 HEX DRIVERS
BINARY COUNTER LATCH 74LS197 E1
28.0MHZ CRYSTAL OSC Y1
4X2 MUX 74LS257 E12
74LS04 E24
74 LS367 E21
74LS00 E67
74LS32 E33
74LS14 E32
74LS02 E17
74LS32 E6
74LS00 E49
74LS86 E200
74 LS74 E68
9643A E10
74LS00 E29
R3 1Kohm 1%
R19 10Kohm
R26 4.7K
R28 220ohm
R5 100Kohm
D1 D664
C1 47uF 10V +50% -10%
C25 470nF 25V
CA04 RESET H
CA04 RESET L
CA04 COUT H
SG1 L
CA06 TDRCV L
CA04 ENIIRD L
CA04 IDS H
CA06 IDS L
CA04 DIDO L
CA02 TDIN H
CA02 TDOUT H
CA04 TPUP L
CA04 TPUP H
CA04 PUP H
CA03 GPFAIL H
PDRAM L
CA01 LDCLO L
CA03 QDCLO L
CA04 LDCLO H
CA04 BCLK250 H
CA04 CLK285 H
CA09 CLK250 H
CA04 CLK571 H
CA04 CLK142 H
CA04 CLK71 H
CA04 CLK35 H
READY H
CA06 ZAK L
CA04 CAS L
CA04 RAS L
CA04 PI H
CA04 R/WLB L
CA04 R/WHB L
CA04 SEL0 H
CA04 SEL1 H
CA06 IREAD L
CA06 DMG L
CA05 CSDTC L
CA04 DMARQ L
CA04 SQB H
CA04 ZBW L
CA04 ZCIEI H
ZCIORQ L
CA06 ZWRITE L
CA06 CSWTDC L
CA06 ZAS L
CA06 ZDS L
CA10 RQPB L
CA11 DREQB L
CA11 DREQA L
EOP L
M47
M48
M49
M50
M51
+5.0V
+3VA
+3VD



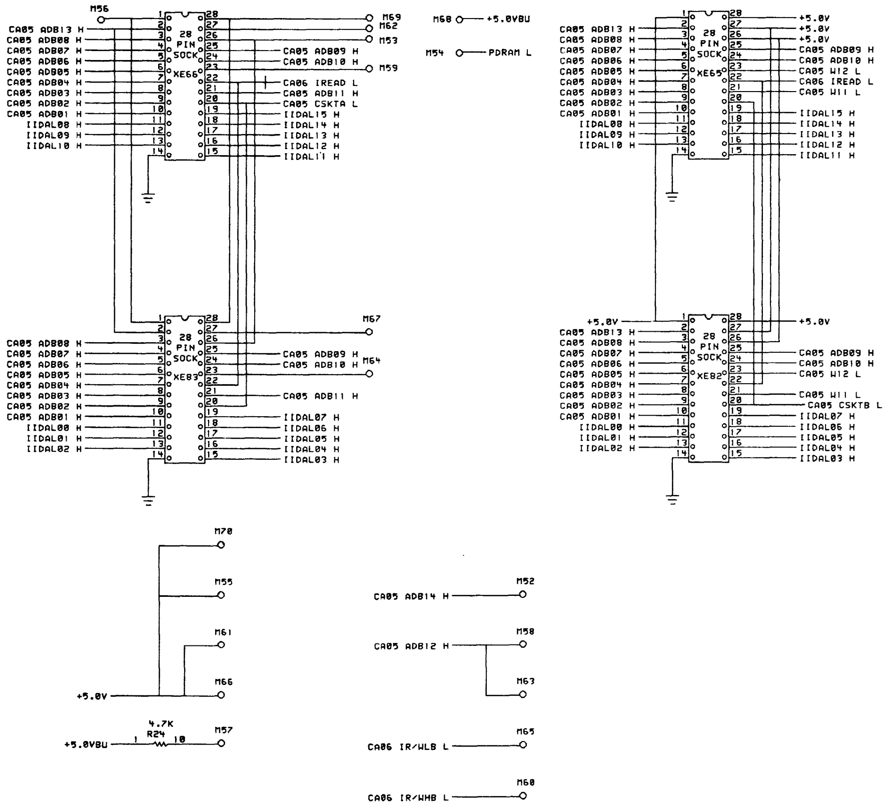
DRAWING 5 OF 16



DRAWING 6 OF 16

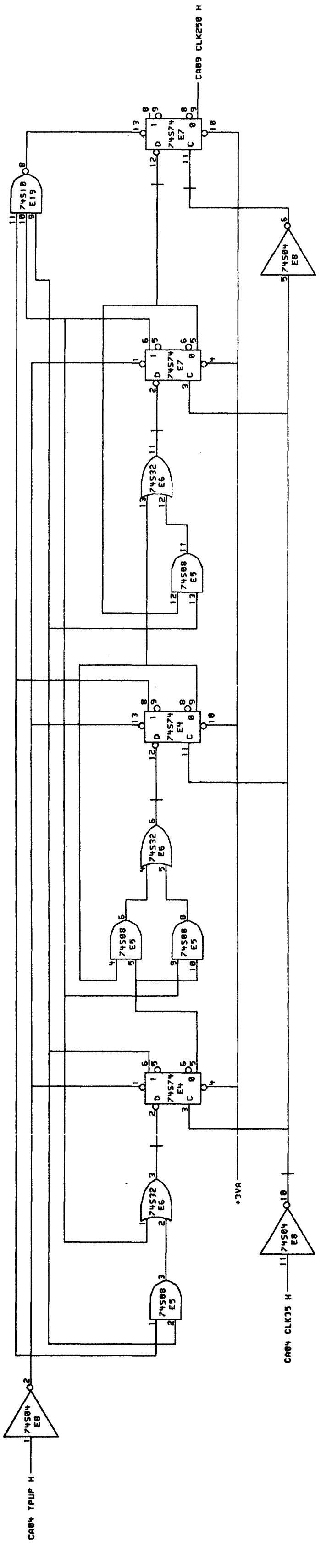


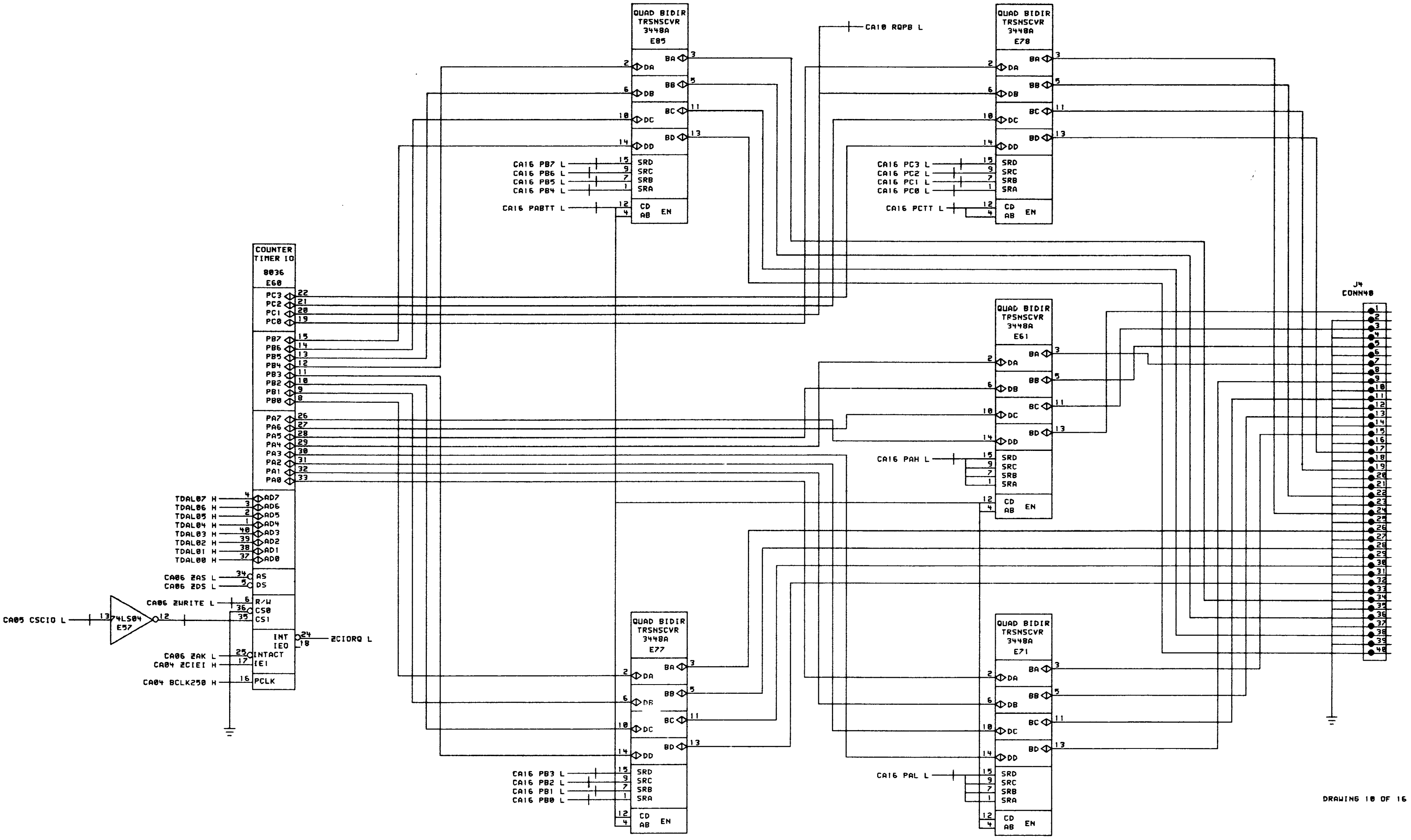
DRAWING 7 OF 16



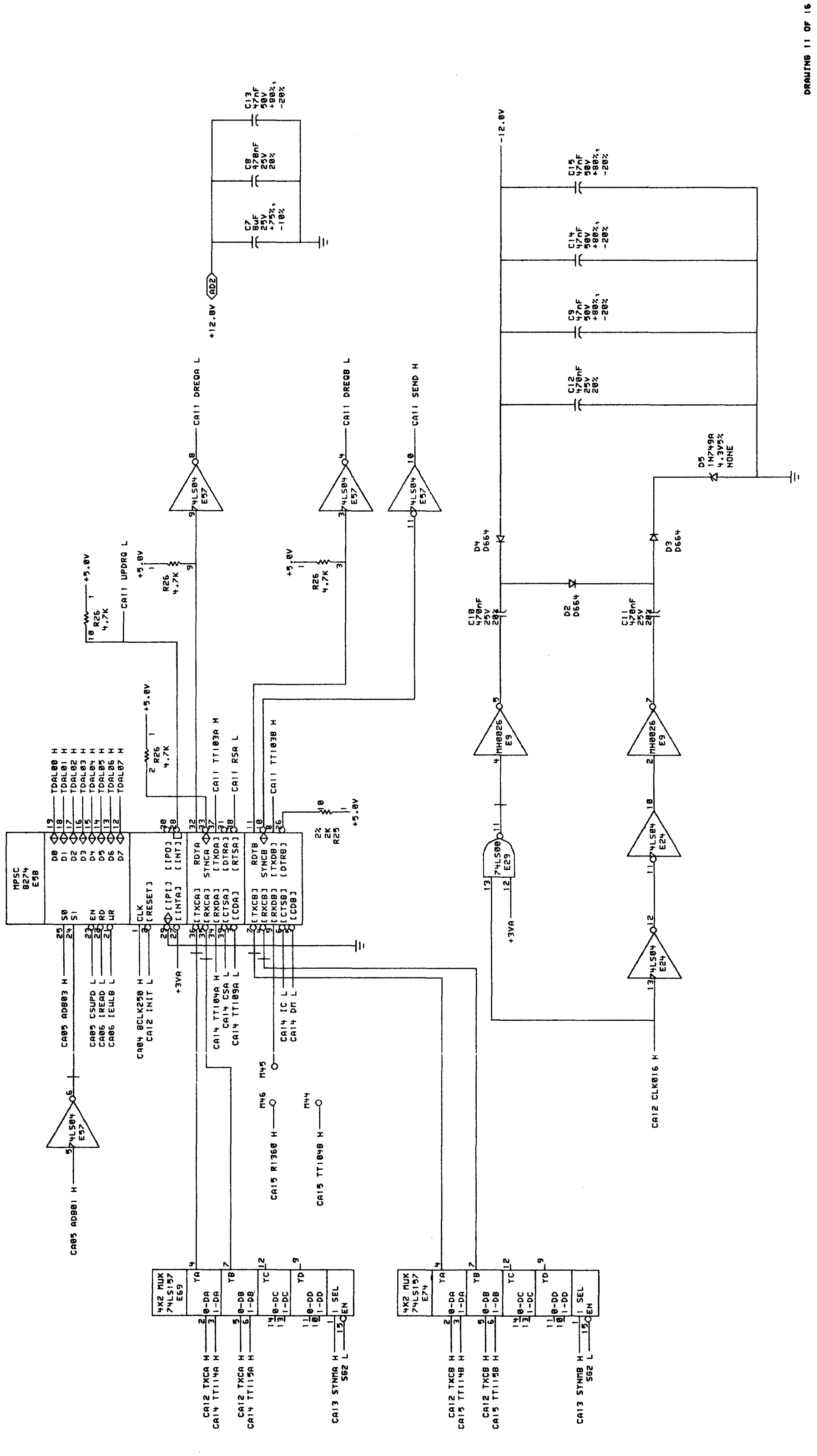
DRAWING 8 OF 16

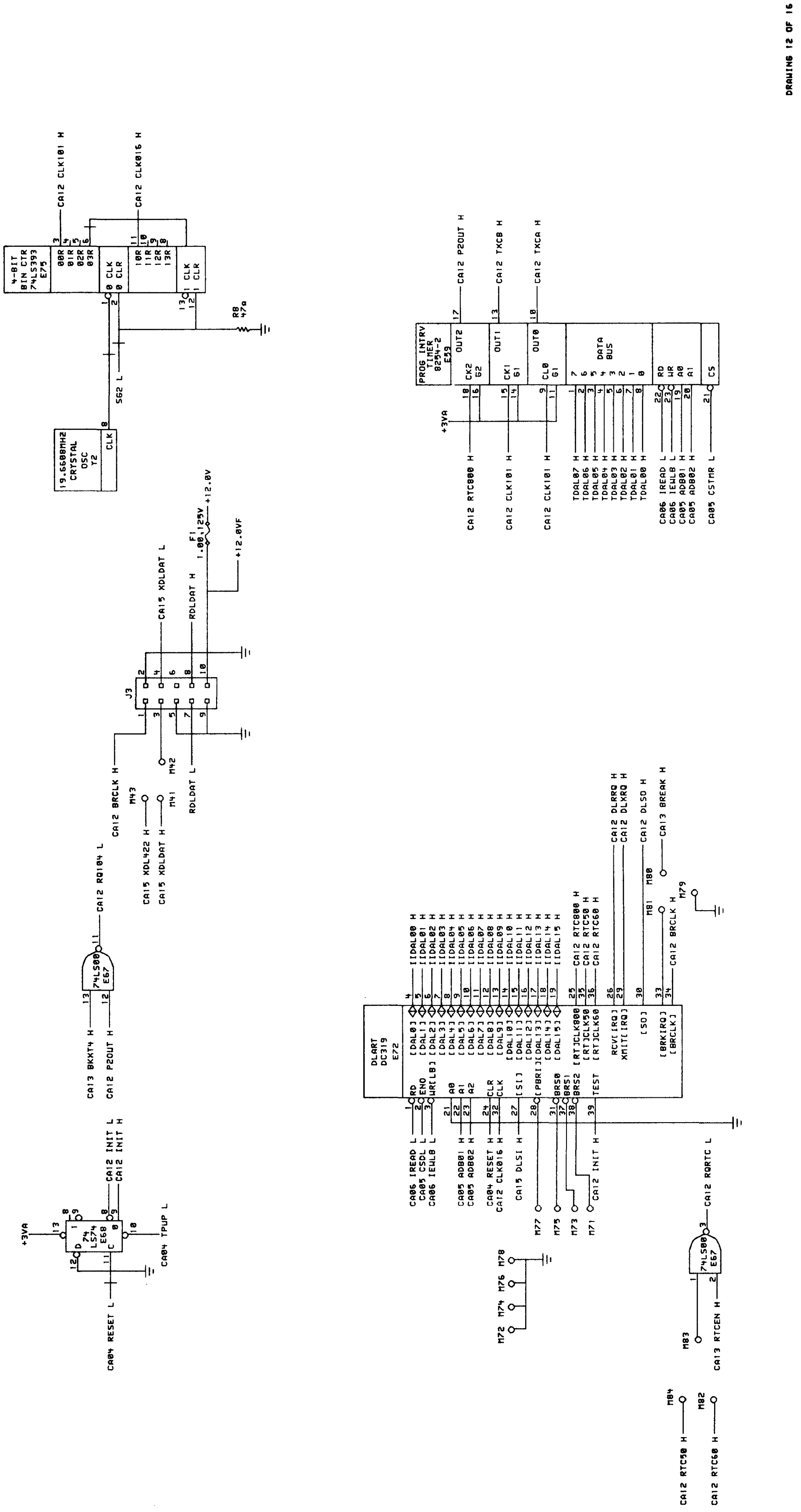
DRAWING 9 OF 16

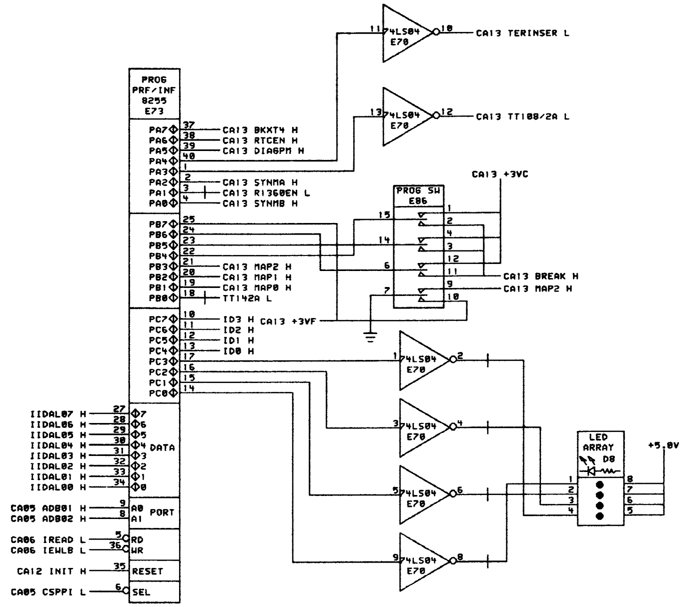




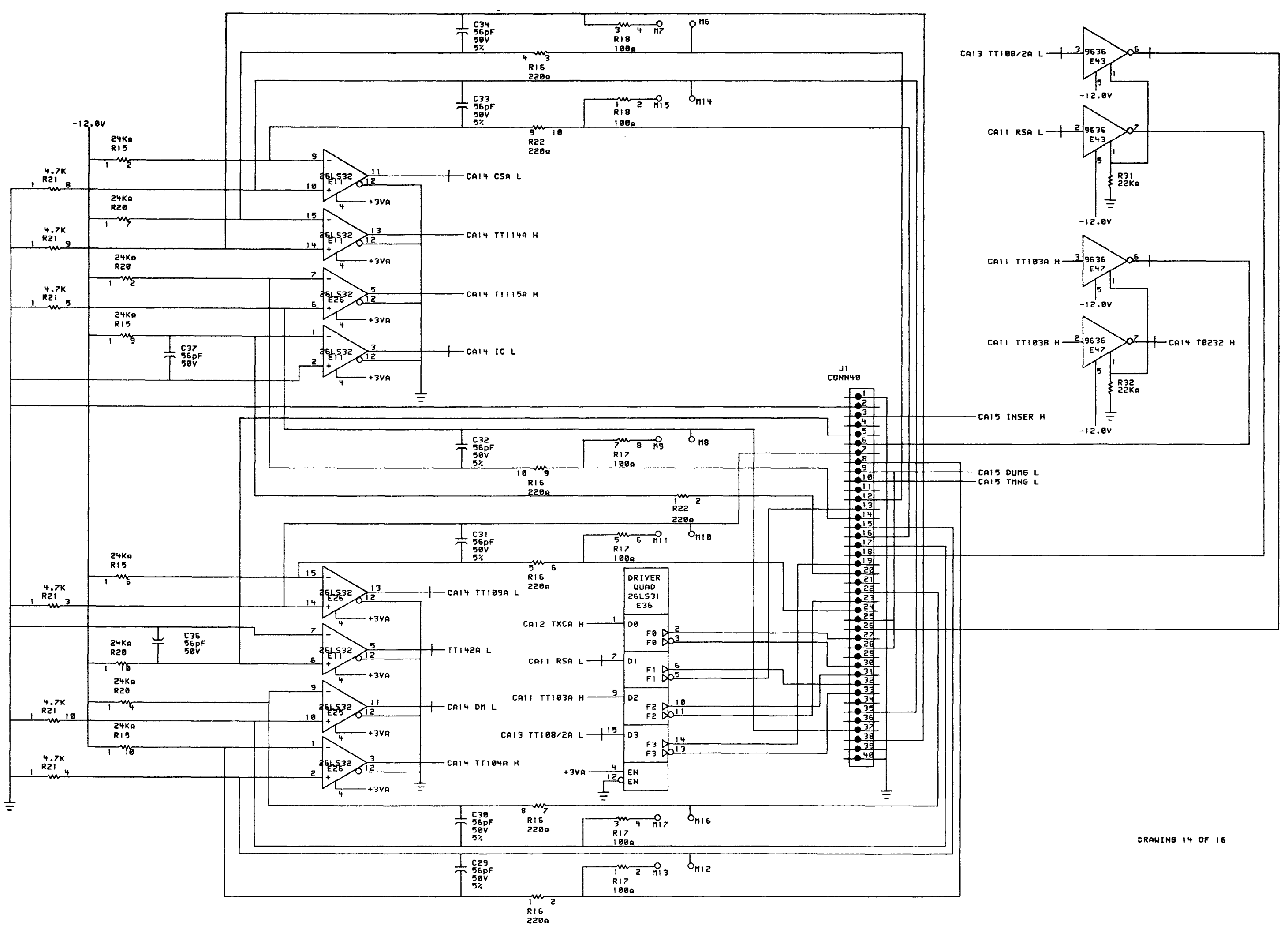
DRAWING 10 OF 16


DRAWING 11 OF 16
MPSC
8274
E58
74LS04
E57
CA05 AD801 H
CA05 AD803 H
CA05 CSUPD L
CA06 IREAD L
CA06 IEWLB L
CA04 BCLK250 H
CA12 INIT L
TDAL00 H
TDAL01 H
TDAL02 H
TDAL03 H
TDAL04 H
TDAL05 H
TDAL06 H
TDAL07 H
R26
4.7K
+5.0V
CA11 UPDRQ L
CA11 DREQA L
CA11 DREQB L
CA11 SEND H
CA11 TT103A H
CA11 RSA L
CA11 TT103B H
2%
2K
R25
+3VA
CA14 TT104A H
CA14 CSA L
CA14 TT109A L
CA14 IC L
CA14 DM L
M45
M46
M44
CA15 RI360 H
CA15 TT104B H
4X2 MUX
74LS157
E69
E74
CA12 TXCA H
CA14 TT114A H
CA14 TT115A H
CA13 SYNMA H
SG2 L
CA12 TXCB H
CA15 TT114B H
CA15 TT115B H
CA13 SYNMB H
74LS00
E29
74LS04
E24
MH0026
E9
CA12 CLK016 H
C10
470nF
25V
20%
C11
470nF
25V
20%
D2
D664
D3
D664
D4
D664
D5
1N749A
4.3V5%
NONE
C12
470nF
25V
20%
C9
47nF
50V
+80%,
-20%
C14
47nF
50V
+80%,
-20%
C15
47nF
50V
+80%,
-20%
-12.0V
+12.0V
AD2
C7
8uF
25V
+75%,
-10%
C8
470nF
25V
20%
C13
47nF
50V
+80%,
-20%

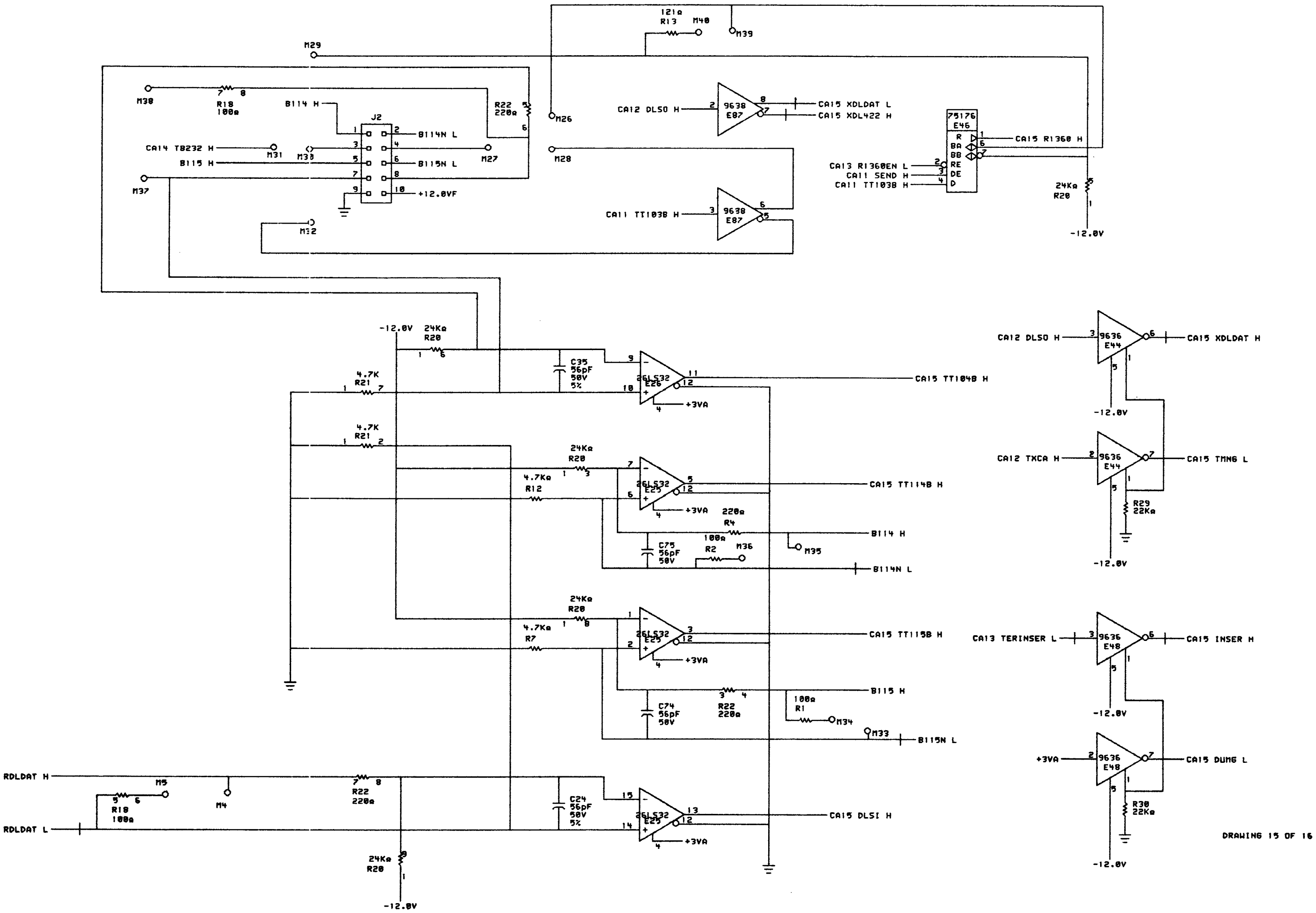

DRAWING 12 OF 16
4-BIT BIN CTR 74LS393 E75
19.6608MHZ CRYSTAL OSC Y2
CA12 CLK101 H
CA12 CLK016 H
SG2 L
R8 47Ω
PROG INTRV TIMER 8254-2 E59
CA12 P2OUT H
CA12 TXCB H
CA12 TXCA H
DATA BUS
CA12 RTC800 H
CA12 CLK101 H
+3VA
TDAL07 H
TDAL06 H
TDAL05 H
TDAL04 H
TDAL03 H
TDAL02 H
TDAL01 H
TDAL00 H
CA06 IREAD L
CA06 IEWLB L
CA05 AD801 H
CA05 AD802 H
CA05 CSTMR L
J3
CA15 XDLDAT L
RDLDAT H
F1 1.00,125V
+12.0V
+12.0VF
CA12 BRCLK H
M43
M42
M41
CA15 XDL422 H
CA15 XDLDAT H
RDLDAT L
CA12 RQ104 L
74LS00 E67
CA13 BKXT4 H
CA12 P2OUT H
+3VA
74 LS74 E68
CA12 INIT L
CA12 INIT H
CA04 RESET L
CA04 TPUP L
DLART DC319 E72
CA06 IREAD L
CA05 CSDL L
CA06 IEWLB L
CA05 AD801 H
CA05 AD802 H
CA04 RESET H
CA12 CLK016 H
CA15 DLSI H
M77
M75
M73
M71
CA12 INIT H
M72 M74 M76 M78
[DAL0]–[DAL15]
[IDAL00 H]–[IDAL15 H]
CA12 RTC800 H
CA12 RTC50 H
CA12 RTC60 H
CA12 DLRRQ H
CA12 DLXRQ H
CA12 DLSO H
CA13 BREAK H
M80
M81
M79
CA12 BRCLK H
74LS00 E67
CA12 RQRTC L
M83
CA13 RTCEN H
M84
M82
CA12 RTC50 H
CA12 RTC60 H



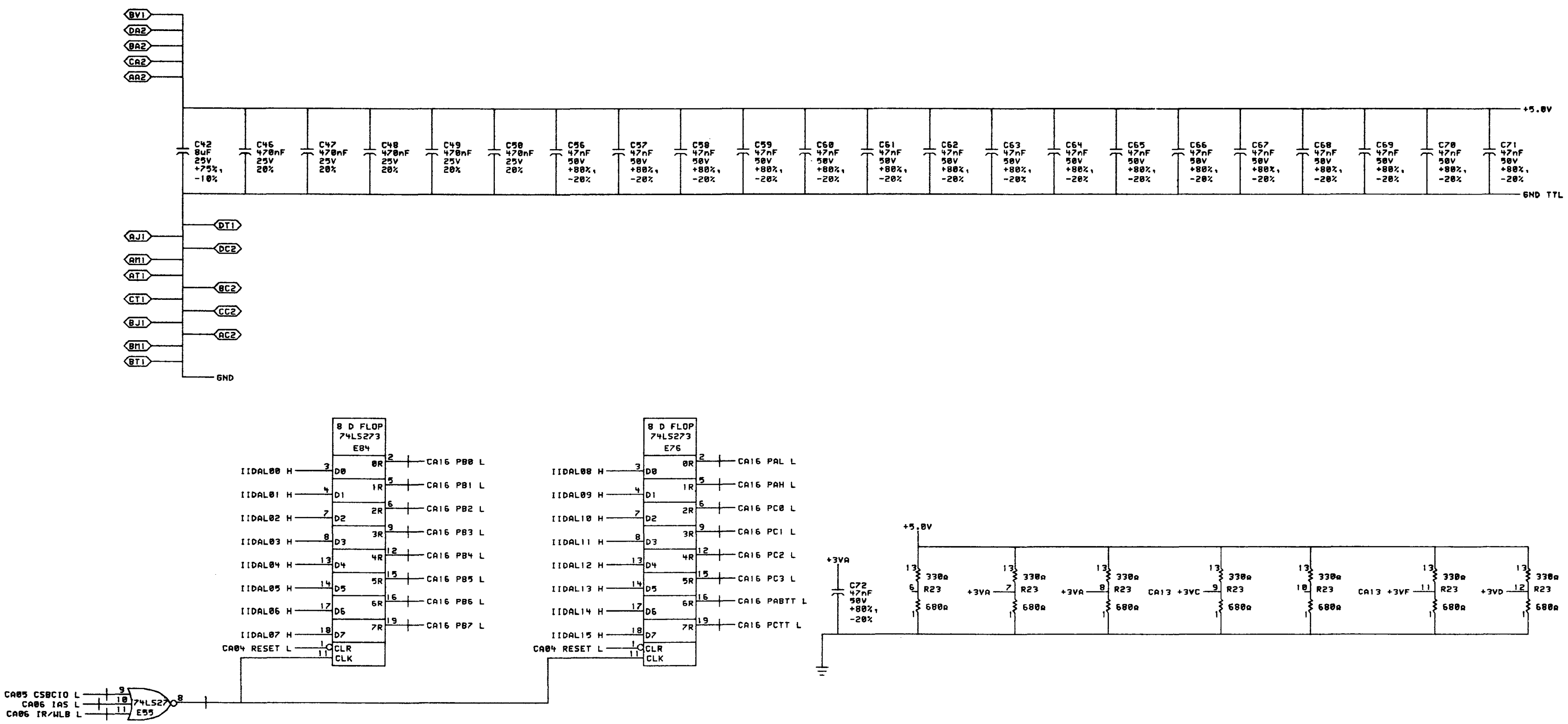
DRAWING 13 OF 16



DRAWING 14 OF 16



DRAWING 15 OF 16



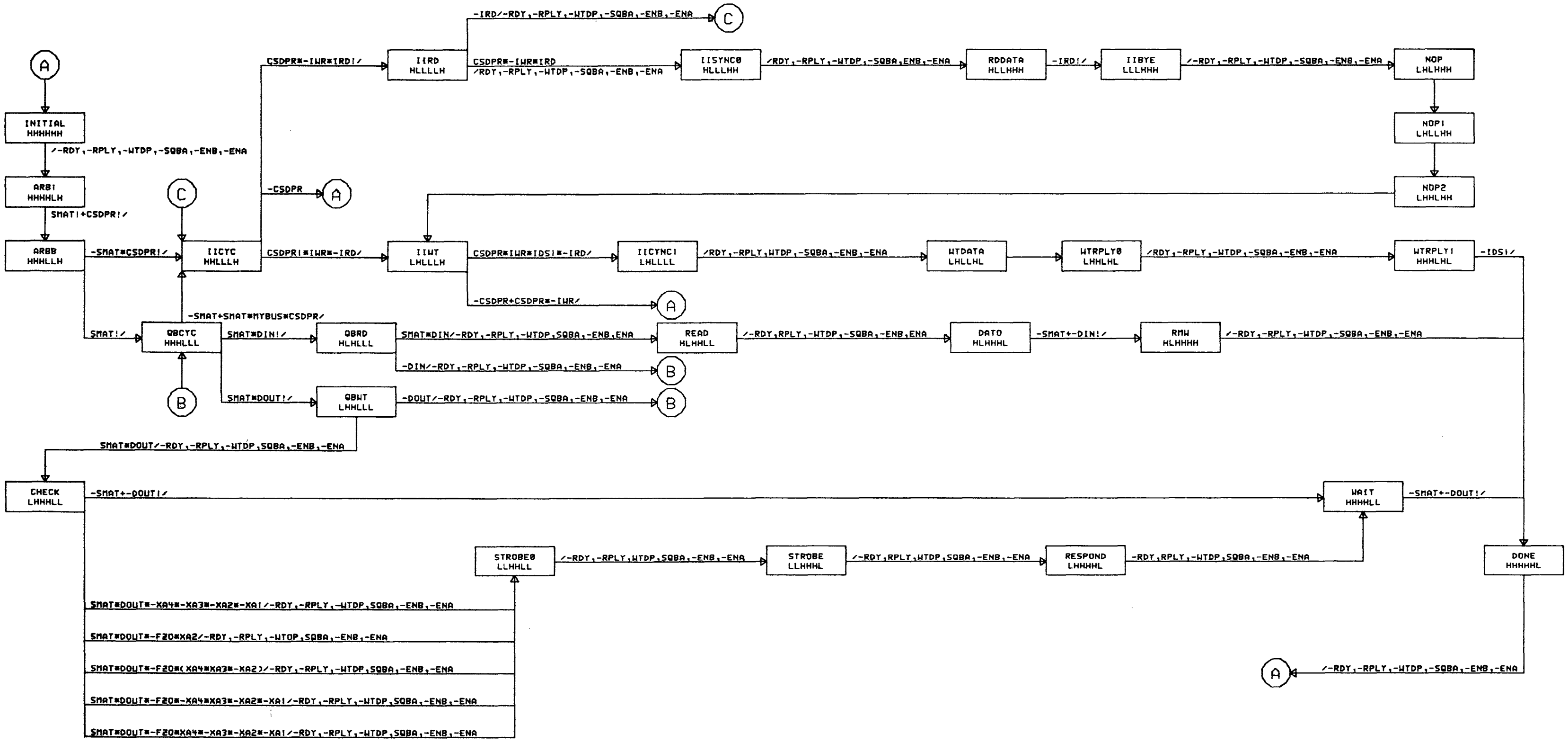
DRAWING 16 OF 16

```
PAGE 01   CA01 BID0 L
          CA01 BID1 L
          CA01 BID2 L
          CA01 BID3 L
          CA01 LDCLO L
          CA01 MATCH H
          CA01 SBCOP H
          CA01 SBCOP L

PAGE 02   CA02 CLK142 L
          CA02 MYBUS H
          CA02 RDIN L
          CA02 RDIN H
          CA02 RDOUT H
          CA02 RINIT H
          CA02 RINIT L
          CA02 RRPLY H
          CA02 RSYNC H
          CA02 TADDR H
          CA02 TDATA L
          CA02 TDIN H
          CA02 TDOUT H
          CA02 TSYNC H
          CA02 TVECTOR L
          CA02 VECT H

PAGE 03   CA03 GPFAIL H
          CA03 IAS H
          CA03 PFAIL H
          CA03 QBER H
          CA03 QDCLO H
          CA03 QDCLO L
          CA03 RHALT H
          CA03 SMATCH L
          CA03 SMATCH H
          CA03 TADDR L
          CA03 TBS7 H
          CA03 TDATA H
          CA03 TWTBT L
          CA03 XAD01 H
          CA03 XAD02 H
          CA03 XAD03 H
          CA03 XAD04 H
          CA03 XDAL16 H
          CA03 XDAL17 H

PAGE 04   CA04 BCLK250 H
          CA04 CAS L
          CA04 CLK142 H
          CA04 CLK285 H
          CA04 CLK35 H
          CA04 CLK571 H
          CA04 CLK71 H
          CA04 COUT H
          CA04 DIDO L
          CA04 DMARQ L
          CA04 EDAL16 H
          CA04 EDAL17 H
          CA04 EDAL18 H
          CA04 EDAL19 H
          CA04 EDAL20 H
          CA04 EDAL21 H
          CA04 ENIIRD L
          CA04 IDS H
          CA04 LDCLO H
          CA04 PI H
          CA04 PUP H
          CA04 QBIOP H
          CA04 R/WHB L
          CA04 R/WLB L
          CA04 RAS L
          CA04 RESET H
          CA04 RESET L
          CA04 SEL0 H
          CA04 SEL1 H
          CA04 SQB H
          CA04 TDAL08 H
          CA04 TDAL09 H
          CA04 TDAL10 H
          CA04 TDAL11 H
          CA04 TDAL12 H
          CA04 TDAL13 H
          CA04 TDAL14 H
          CA04 TDAL15 H
          CA04 TPUP L
          CA04 TPUP H
          CA04 ZBW L
          CA04 ZCIEI H

PAGE 05   CA05 ADB01 H
          CA05 ADB02 H
          CA05 ADB03 H
          CA05 ADB04 H
          CA05 ADB05 H
          CA05 ADB06 H
          CA05 ADB07 H
          CA05 ADB08 H
          CA05 ADB09 H
          CA05 ADB10 H
          CA05 ADB11 H
          CA05 ADB12 H
          CA05 ADB13 H
          CA05 ADB14 H
          CA05 ADB15 H
          CA05 ADIAGPM H
          CA05 CSBCIO L
          CA05 CSCIO L
          CA05 CSDL L
          CA05 CSDPR L
          CA05 CSDTC L
          CA05 CSIIDAL L
          CA05 CSKTA L
          CA05 CSKTB L
          CA05 CSLIP1 L
          CA05 CSLIP4 L
          CA05 CSPPI L
          CA05 CSQIKX L
          CA05 CSRAM1 L
          CA05 CSRAM2 L
          CA05 CSTMR L
          CA05 CSUPD L
          CA05 GRQTM L
          CA05 RQTM L
          CA05 SIIDAL L
          CA05 W11 L
          CA05 W12 L

PAGE 06   CA06 AK120 L
          CA06 AK124 L
          CA06 AK130 L
          CA06 AK134 L
          CA06 BRQ140 H
          CA06 CLK71 L
          CA06 CLRRS H
          CA06 CSWTDTC L
          CA06 DC10RP H
          CA06 DC3RQ H
          CA06 DE646 L
          CA06 DISDE H
          CA06 DMG L
          CA06 DPRDY L
          CA06 DPRHIT H
          CA06 ENAPT L
          CA06 ENBPT L
          CA06 F20 L
          CA06 IAS L
          CA06 IDS L
          CA06 IEWLB L
          CA06 IR/WHB L
          CA06 IR/WLB L
          CA06 IREAD L
          CA06 IWRITE L
          CA06 QRESETOC L
          CA06 RQ120 H
          CA06 RQ124 H
          CA06 RQ130 H
          CA06 RQ134 H
          CA06 RQDC10 H
          CA06 SQBAD L
          CA06 TDRCV L
          CA06 TRPLY L
          CA06 WTDPR L
          CA06 ZAK L
          CA06 ZAS L
          CA06 ZDS L
          CA06 ZWRITE L

PAGE 09   CA09 CLK250 H

PAGE 10   CA10 RQPB L

PAGE 11   CA11 DREQA L
          CA11 DREQB L
          CA11 RSA L
          CA11 SEND H
          CA11 TT103A H
          CA11 TT103B H
          CA11 UPDRQ L

PAGE 12   CA12 BRCLK H
          CA12 CLK016 H
          CA12 CLK101 H
          CA12 DLRRQ H
          CA12 DLSO H
          CA12 DLXRQ H
          CA12 INIT L
          CA12 INIT H
          CA12 P2OUT H
          CA12 RQ104 L
          CA12 RQRTC L
          CA12 RTC50 H
          CA12 RTC60 H
          CA12 RTC800 H
          CA12 TXCA H
          CA12 TXCB H

PAGE 13   CA13 +3VC
          CA13 +3VF
          CA13 BKXT4 H
          CA13 BREAK H
          CA13 DIAGPM H
          CA13 MAP0 H
          CA13 MAP1 H
          CA13 MAP2 H
          CA13 R1360EN L
          CA13 RTCEN H
          CA13 SYNMA H
          CA13 SYNMB H
          CA13 TERINSER L
          CA13 TT108/2A L

PAGE 14   CA14 CSA L
          CA14 DM L
          CA14 IC L
          CA14 TB232 H
          CA14 TT104A H
          CA14 TT109A L
          CA14 TT114A H
          CA14 TT115A H

PAGE 15   CA15 DLSI H
          CA15 DUMG L
          CA15 INSER H
          CA15 R1360 H
          CA15 TMNG L
          CA15 TT104B H
          CA15 TT114B H
          CA15 TT115B H
          CA15 XDL422 H
          CA15 XDLDAT L
          CA15 XDLDAT H

PAGE 16   CA16 PABTT L
          CA16 PAH L
          CA16 PAL L
          CA16 PB0 L
          CA16 PB1 L
          CA16 PB2 L
          CA16 PB3 L
          CA16 PB4 L
          CA16 PB5 L
          CA16 PB6 L
          CA16 PB7 L
          CA16 PC0 L
          CA16 PC1 L
          CA16 PC2 L
          CA16 PC3 L
          CA16 PCTT L
```

NON-SOURCED SIGNALS

```
+12.0VF
+3VA
+3VD
+5.0VBB
+5.0VBU
AI0 L
AI1 L
AI2 L
AI3 L
AI4 L
AI5 L
AI6 L
AI7 L
B114 H
B114N L
B115 H
B115N L
BBS7 L
BCM2 L
BCN2 L
BCR2 L
BCS2 L
BDAL00 L
BDAL01 L
BDAL02 L
BDAL03 L
BDAL04 L
BDAL05 L
BDAL06 L
BDAL07 L
BDAL08 L
BDAL09 L
BDAL10 L
BDAL11 L
BDAL12 L
BDAL13 L
BDAL14 L
BDAL15 L
BDAL16 L
BDAL17 L
BDAL18 L
BDAL19 L
BDAL20 L
BDAL21 L
BDCOK H
BDIN L
BDMGI L
BDMGO L
BDMR L
BDOUT L
BHALT L
BIAKI L
BIAKO L
BINIT L
BIRQ4 L
BPOK H
BRPLY L
BSACK L
BSYNC L
BWTBT L
DLRC L
EOP L
ID0 H
ID1 H
ID2 H
ID3 H
IIDAL00 H
IIDAL01 H
IIDAL02 H
IIDAL03 H
IIDAL04 H
IIDAL05 H
IIDAL06 H
IIDAL07 H
IIDAL08 H
IIDAL09 H
IIDAL10 H
IIDAL11 H
IIDAL12 H
IIDAL13 H
IIDAL14 H
IIDAL15 H
PDRAM L
RDLDAT L
RDLDAT H
READY H
SG1 L
SG2 L
TDAL00 H
TDAL01 H
TDAL02 H
TDAL03 H
TDAL04 H
TDAL05 H
TDAL06 H
TDAL07 H
TT142A L
XDAL00 H
XDAL01 H
XDAL02 H
XDAL03 H
XDAL04 H
XDAL05 H
XDAL06 H
XDAL07 H
XDAL08 H
XDAL09 H
XDAL10 H
XDAL11 H
XDAL12 H
XDAL13 H
XDAL14 H
XDAL15 H
XDAL18 H
XDAL19 H
XDAL20 H
XDAL21 H
ZCIORQ L
```

IF CONDITION/THEN OPERATION


A
INITIAL
HHHHHH
/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
ARB1
HHHHLH
SMAT1+CSDPR!/
ARBB
HHHLLH
-SMAT*CSDPR!/
C
IICYC
HHLLLH
CSDPR*-IWR*IRD!/
IIRD
HLLLLH
-IRD/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
C
CSDPR*-IWR*IRD
/RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
IISYNC0
HLLLHH
/RDY,-RPLY,-WTDP,-SQBA,ENB,-ENA
RDDATA
HLLHHH
-IRD!/
IIBYE
LLLHHH
/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
NOP
LHLHHH
NOP1
LHLLHH
NOP2
LHHLHH
-CSDPR
A
CSDPR1*IWR*-IRD/
IIWT
LHLLLH
CSDPR*IWR*IDS1*-IRD/
IICYNC1
LHLLLL
/RDY,-RPLY,WTDP,-SQBA,-ENB,-ENA
WTDATA
LHLLHL
WTRPLY0
LHHLHL
/RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
WTRPLY1
HHHLHL
-IDS1/
-CSDPR+CSDPR*-IWR/
A
SMAT!/
QBCYC
HHHLLL
-SMAT+SMAT*MYBUS*CSDPR/
SMAT*DIN!/
QBRD
HLHLLL
SMAT*DIN/-RDY,-RPLY,-WTDP,SQBA,-ENB,ENA
READ
HLHHLL
/-RDY,RPLY,-WTDP,-SQBA,-ENB,ENA
DATO
HLHHHL
-SMAT+-DIN!/
RMW
HLHHHH
/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
-DIN/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
B
B
SMAT*DOUT!/
QBWT
LHHLLL
-DOUT/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA
B
SMAT*DOUT/-RDY,-RPLY,-WTDP,SQBA,-ENB,-ENA
CHECK
LHHHLL
-SMAT+-DOUT!/
WAIT
HHHHLL
-SMAT+-DOUT!/
DONE
HHHHHL
STROBE0
LLHHLL
/-RDY,-RPLY,WTDP,SQBA,-ENB,-ENA
STROBE
LLHHHL
/-RDY,RPLY,WTDP,SQBA,-ENB,-ENA
RESPOND
LHHHHL
-RDY,RPLY,-WTDP,SQBA,-ENB,-ENA
SMAT*DOUT*-XA4*-XA3*-XA2*-XA1/-RDY,-RPLY,-WTDP,SQBA,-ENB,-ENA
SMAT*DOUT*-F20*XA2/-RDY,-RPLY,-WTOP,SQBA,-ENB,-ENA
SMAT*DOUT*-F20*(XA4*XA3*-XA2)/-RDY,-RPLY,-WTDP,SQBA,-ENB,-ENA
SMAT*DOUT*-F20*-XA4*XA3*-XA2*-XA1/-RDY,-RPLY,-WTDP,SQBA,-ENB,-ENA
SMAT*DOUT*-F20*XA4*-XA3*-XA2*-XA1/-RDY,-RPLY,-WTDP,SQBA,-ENB,-ENA
A
/-RDY,-RPLY,-WTDP,-SQBA,-ENB,-ENA

DUAL PORT RAM STATE MACHINE

APPENDIX E
ADDRESS ENVIRONMENT

Table E-1 lists all KXT11-CA locally addressable registers. The register addresses listed under the two-port register file description are addressable by the Q-Bus. The Q-Bus address is the local address plus the CPU ID offset.

Table E-1 KXT11-CA Address Assignments

Local Address	Level	Vectors	Description
177566–177560	4i	60,64	SLU1 console DL registers
177532	5i, Q-Bus	130 level 4	Q-Bus interrupt register (QIR)
177530 177526	6i		KXTCSR D Mode control word for KXTCSR A, B and C. Do not alter
177524–177520	6i	104 (P CNTR)	KXTCSR A, KXTCSR B, KXTCSR C
177140			Parallel I/O Buffer control register
177136–177000	4e	Read-in programmable	Parallel I/O and timers registers
173736–175700	4i	70	SLU2 synchronous/asynchronous serial I/O channel A, channel B, and counter/timer registers
175036–175032 175030	5i	134w	Two-Port register file:
175026–175022 175020	5i	124w	User control secondary
175016–175012 175010	5i	120w	User control primary
175006–175002	(nonmaskable)	173004w	System control
174536–174400	4e	Read-in programmable	DMA controller registers
173000 173004			T-11 Power-Up start address T-11 Restart address
173777–164000			Two 28-pin system PROM socket sites for OCT/boot/self-test/diagnostics
163777–160000			1 K word overlay PROM for diagnostics
15777–0000000			Two 28-pin user sockets for RAM or PROM and 16 KW static system RAM.

i = T-11 internal
e = T-11 external
w = Q-Bus write induced

On an interrupt level, internal (i) interrupts have priority over external (e) interrupts.

APPENDIX F
LOOPBACK CONNECTORS

Three different types of loopback connectors are needed if certain built-in selftests are selected to be performed. Users can build their own loopback connectors that are compatible with the built-in selftest. Two are 10-pin loopback connectors for testing via I/O port J2 (SLU2 channel B) and via I/O port J3 (SLU1). See Figure F-1 for loopback connector details. A 40-pin loopback connector is for testing via I/O port J1 (SLU2 channel A). This loopback set up 10 has switches enabling the testing of either RS423 or RS422 connector operation. See Figure F-2 for details. One 40-pin loopback connector is used for testing via I/O port J4 (parallel port). See Figure F-3 for details.

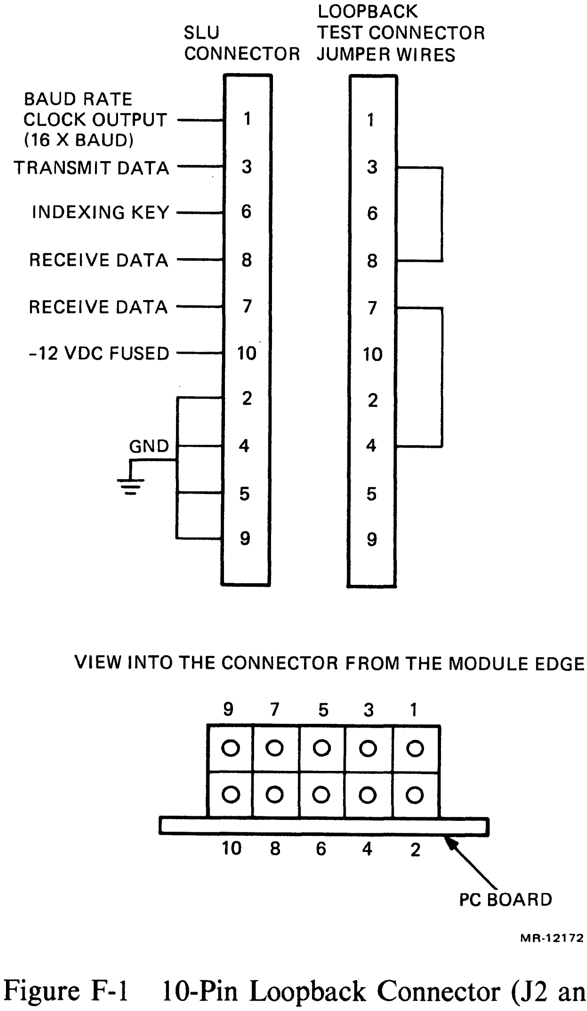


Figure F-1 10-Pin Loopback Connector (J2 and J3)

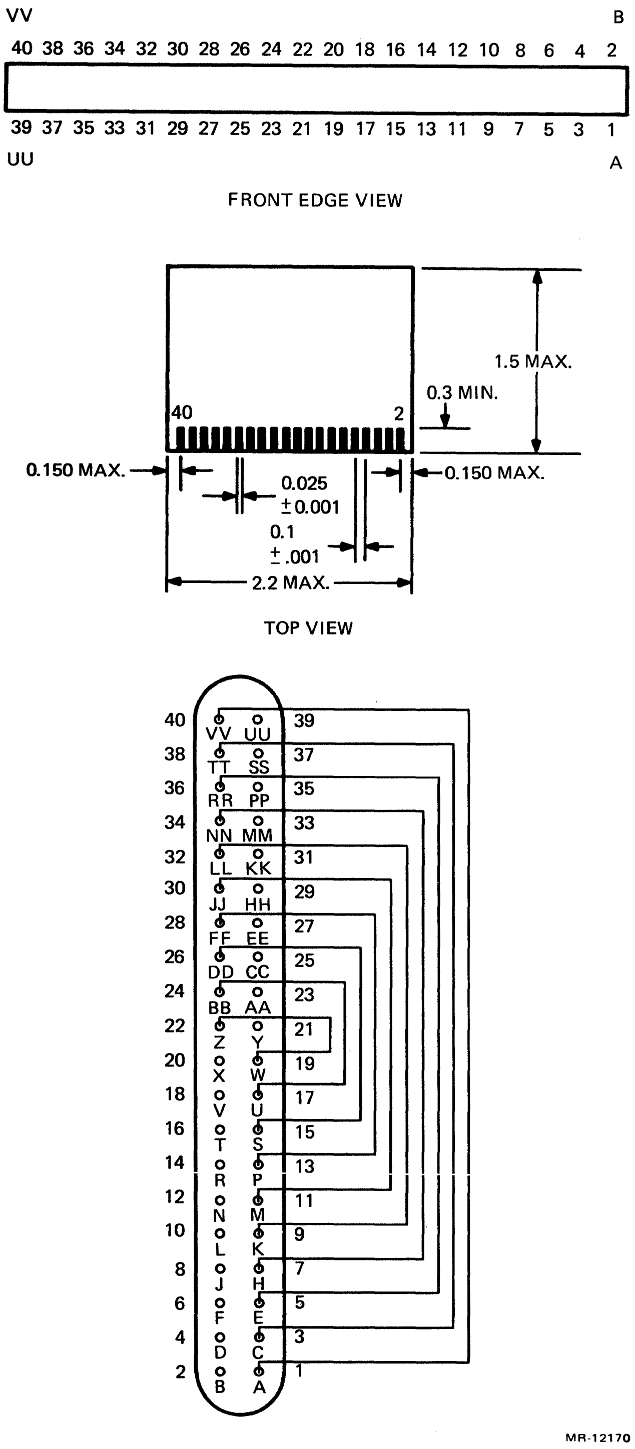
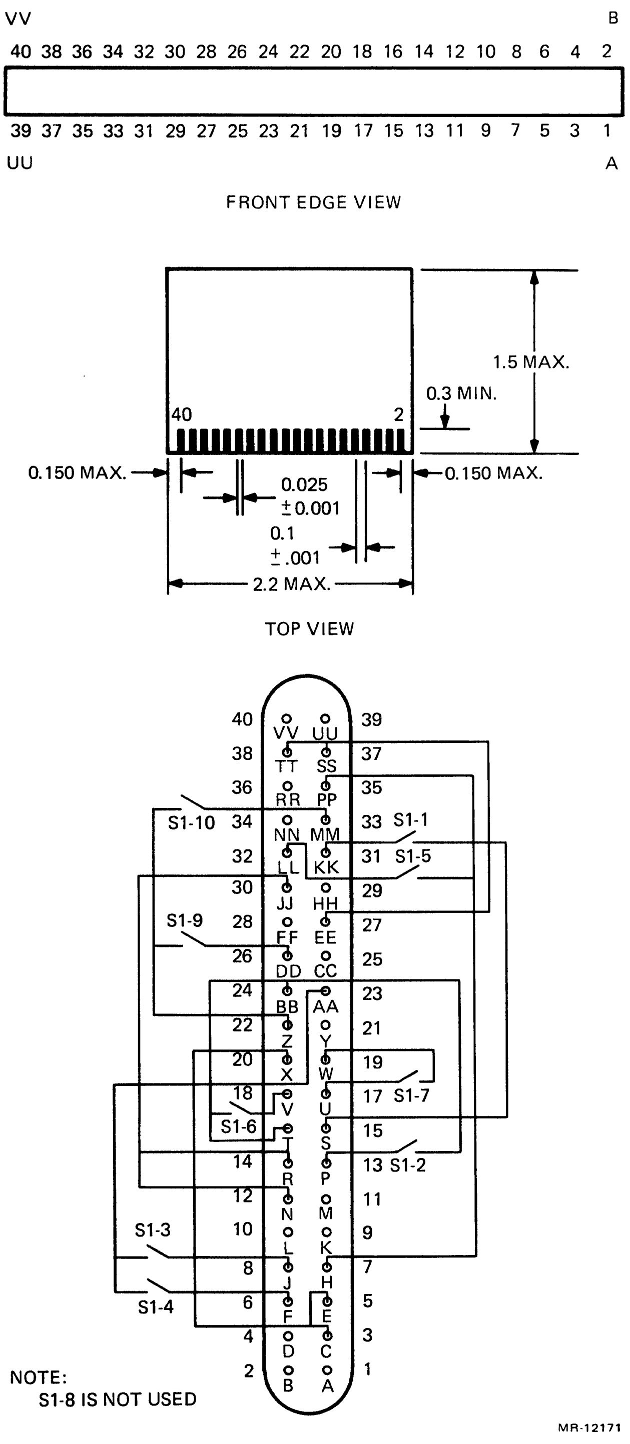


Figure F-2 40-Pin Loopback Connector (J1)



NOTE:
S1-8 IS NOT USED

Figure F-3 40-Pin Loopback Connector (J4)

APPENDIX G
GLOSSARY

AD01–AD15 – A 15-bit on-board address bus used to address memory and peripheral devices. Generated by two 8-bit latches that are loaded from the TDAL bus. See also BBS7.

AI0–AI7 – Input lines used by the microprocessor for interrupts and DMA requests.

ASPI – Microprocessor transaction that allows the microprocessor to recognize and accept pending interrupts or DMA requests.

Autobaud – Self-adjusting baud rates for SLU1 only. Implemented by firmware in the optional Macro-ODT ROM.

BBS7 – Q-Bus signal indicating that the device addressed is in the I/O page.

BDAL 0–15 – Multiplexed data and address lines of the Q-Bus connected through the backplane.

BDCOK – Q-Bus signal that goes high 3 ms after dc power is applied and stays high until 4 ms after dc power is removed.

BDIN – Q-Bus data input strobe.

BDMGI – Q-Bus DMA grant signal derived from the BDMGO bus pin. It is daisy chained through each SBC entering on the BDMGI pin and exiting on the BDMGO pin. It represents the bus grant for the DMA control line.

BDMGO – See BDMGI.

BDMR – DMA request signal from the Q-Bus.

BDOUT – Q-Bus data output strobe.

BEVNT – Q-Bus signal used to generate REVNT. Can be used to initiate an interrupt.

BHALT – Q-Bus halt signal used for a priority 7 interrupt that vectors through location 140.

BINIT – Q-Bus signal used to initialize all the devices on the bus.

BIRQ4 – Q-Bus level 4 priority interrupt request that is used to initiate the internal IRQ4 signal.

BKRQ – Internal control signal initiated by BHALT or BREAK detect from terminal.

BPOK – Q-Bus signal that goes high 70 ms after BDCOK and goes low when ac power is lost.

BREAK – Initiated by pressing the BREAK key. Causes the KXT11-CA SBC to generate BREAK H that is used as an interrupt.

BRPLY – Slave's acknowledge of a Q-Bus cycle.

BSACK – Acknowledges receipt of a DMA grant signal.

BSYNC – Q-Bus cycle control signal.

BWTBT – Q-Bus write byte control signal.

CAS – An output from the microprocessor that acts as data strobe. Used for the read/write, DMA, and ASPI transactions.

Condition codes – The least significant four bits of the processor status word that indicate the results of the last instruction executed.

Configuration – Allows the user to select optional features of the SBC by inserting jumpers.

Control and status register (CSR) – Internal register that allows the program to control and monitor the operation of the SBC.

Control word – The data contained in the control register of the parallel I/O chip that determines the configuration of the parallel I/O interface.

COUT – An output from the microprocessor clock that is asserted once during each microcycle.

CSKTA – The RAM/ROM socket set A chip select strobe.

CSKTB – The RAM/ROM socket set B chip select strobe.

CTMER – Time-out interrupt that has the same effect as HALT.

Cycle slip – This condition exists when the READY input is pulsed while RAS is asserted. It causes the microprocessor to be idle, and no transactions occur.

DATI – Q-Bus transaction that transfers 16 bits of data from the slave to the master.

DATO – Q-Bus transaction that transfers 16 bits of data from the master to the slave.

DATO(B) – Q-Bus transaction that transfers 8 bits of data from the master to the slave.

DMA – Direct memory access for transferring blocks of data without program intervention.

DMA transaction – A microprocessor transaction during which the microprocessor gives up bus mastership to another device for direct transfer of memory data.

DTC – Direct transfer controller for DMA transactions.

EIA RS232-C – Electronics Industries Association serial line interface standard.

EIA RS423 – Electronics Industries Association serial line interface standard.

EOP – End of Process or End of Pass.

Fetch/read – Microprocessor transaction that transfers data from memory or I/O into the microprocessor. The data may be an instruction (fetch) or an operand (read).

Firmware – The programs that reside in the PROM or ROM hardware.

FPLA – Field programmable logic array. Used to decode memory addresses.

HALT – The highest priority interrupt. Causes the microprocessor to go to the restart address and loads the PSW with 340.

Handshaking protocol – The series of events used to establish data transfers.

IAK – Microprocessor transaction to acknowledge an interrupt and secure a vector from an on-board location or from the Q-Bus.

IIDAL 0–15 – Local multiplexed data and address lines used for DMA transfers.

Interrupts – Interruption of the normal program execution to service an external request.

Interrupt protocol – Signal sequence required to initiate and service interrupts.

Interrupt vector – The memory location in which the address of the interrupt service routine is stored.

LSI-11 bus – An asynchronous bus that provides interconnections for LSI-11 type modules.

Maskable – A priority level that can be inhibited by loading the PSW with a higher priority code.

Memory mapping – The partitioning of memory via jumper configurations to determine the on-board portions and the Q-Bus portions of memory.

Microcycle – The time necessary to execute one microinstruction. A transaction may use three or four microcycles.

Mode register – A register used to define the microprocessor's operations structure.

Nibble – The upper or lower half of a byte that consists of four bits.

Nonmaskable – A priority level that is higher than the level selectable by the PSW. An interrupt which cannot be disabled.

NOP – A transaction that produces no useful output. It is used to introduce a delay or wait period.

Parallel I/O – Parallel data interface.

Parallel I/O handshaking – Control signals used to establish parallel data transfers.

PI (priority in) – A microprocessor output signal used to strobe interrupt and DMA requests into the microprocessor.

Power-Fail (PFAIL) – A nonmaskable interrupt caused by a power failure that causes the microprocessor to vector through location 24 to the power-fail routine.

Priority – Bits 5 and 6 of the PSW. Used to define the priority level of the microprocessor.

PSW register – A microprocessor register that contains the processor status word (PSW).

PUP – An input to the microprocessor that controls the power-up sequence. When it is switched from high to low, the microprocessor power-up sequence is initiated.

Q-Bus – See LSI-11 bus definition.

QIR – Interrupt register to the Q-Bus arbiter.

RAM – Random-access memory defined as read/write memory.

RAS – Microprocessor output used as an address strobe in read/write, IAK, and DMA transactions.

READY – Input to the microprocessor that causes cycle slips when pulsed.

Restart address – The address that the microprocessor jumps to when executing a HALT interrupt.

REVNT – See BEVNT.

ROM – Read-only memory that cannot be modified with a normal write operation.

R/WHB – A microprocessor output that is low for high byte write transactions and high for read transactions.

R/WLB – A microprocessor output that is low for low byte write transactions and high for read transactions.

RTI – Return from interrupt instruction.

SBC – Single board computer.

SEL0/SEL1 – Microprocessor outputs used to define the transaction being performed.

Serial I/O – Asynchronous serial line units for the transfer of serial data. SLU1 and SLU2 are two such units used in the KXT11-CA SBC.

Spurious halts – Halt conditions that are not internally created in normal operation.

Stack pointer – The register that contains the address of the last word stored on the stack.

Start address – The address that the microprocessor goes to during power-up.

TDAL 0–15 – Internal on-board bus used for multiplexed data and address lines. See BDAL 0–15.

Trace bit – Bit 4 of the PSW that causes a trap to location 14 after the next instruction is completed.

Transaction – A sequence of microcycles used to complete a designated microprocessor function such as read, write, ASPI, or IAK.

Tristate – Designates that a signal is in a high impedance condition and not a logic “1” or “0”.

Vector address – Memory location the microprocessor accesses for the address of the interrupt service routine during an interrupt.

Wait state – A condition during which the microprocessor performs no useful transactions while waiting for a response or data.

Wakeup circuit – Holds BDCOK negated for 50 ms after dc power has been applied.

INDEX

INDEX (Cont)

Digital Equipment Corporation • Bedford, MA 01730