
DECrouter
200

DECrouter 200

Technical Manual

Order No. EK-DR200-TG-001

January 1987

The *DECrouter 200 Technical Manual* provides general operating instructions, detailed hardware logical functions, and diagnostic software information.

Supersession/Update Information: This is a new manual

digital

EK-DR200-TG-001
First Printing, January 1987

The postage-prepaid Readers Comments form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

This manual was produced by Networks and Communications Publications.

Contents

4 On-Line Debugging Tool (ODT)

5 Functional/Logic Description

6 Hardware Description

Figures

Tables

Preface

The DECrouter 200 Technical Manual provides general operating instructions, detailed hardware logical functions, and diagnostic software information. The DECrouter 200 is also referred to as the router throughout this manual.

Intended Audience

The manual is for use in training, field service, and manufacturing. The depth of technical information requires previous training or experience with Ethernet networks and with Digital VAX-11 or PDP-11 architecture.

Manual Organization

The manual is divided into the following six chapters:

Chapter 1 Introduces the DECrouter 200 and the Ethernet communications system.

Chapter 2 Explains the self-test program diagnostic modes and test sequences.

Chapter 3 Describes how the initialize program handles down-line loading and up-line dumping. Chapter 3 also explains status messages and error messages.

Chapter 4 Explains the on-line debugging tool (ODT) commands and how the commands are used.

Chapter 5 Provides block diagram level and register address-level descriptions of the DECrouter 200 logic.

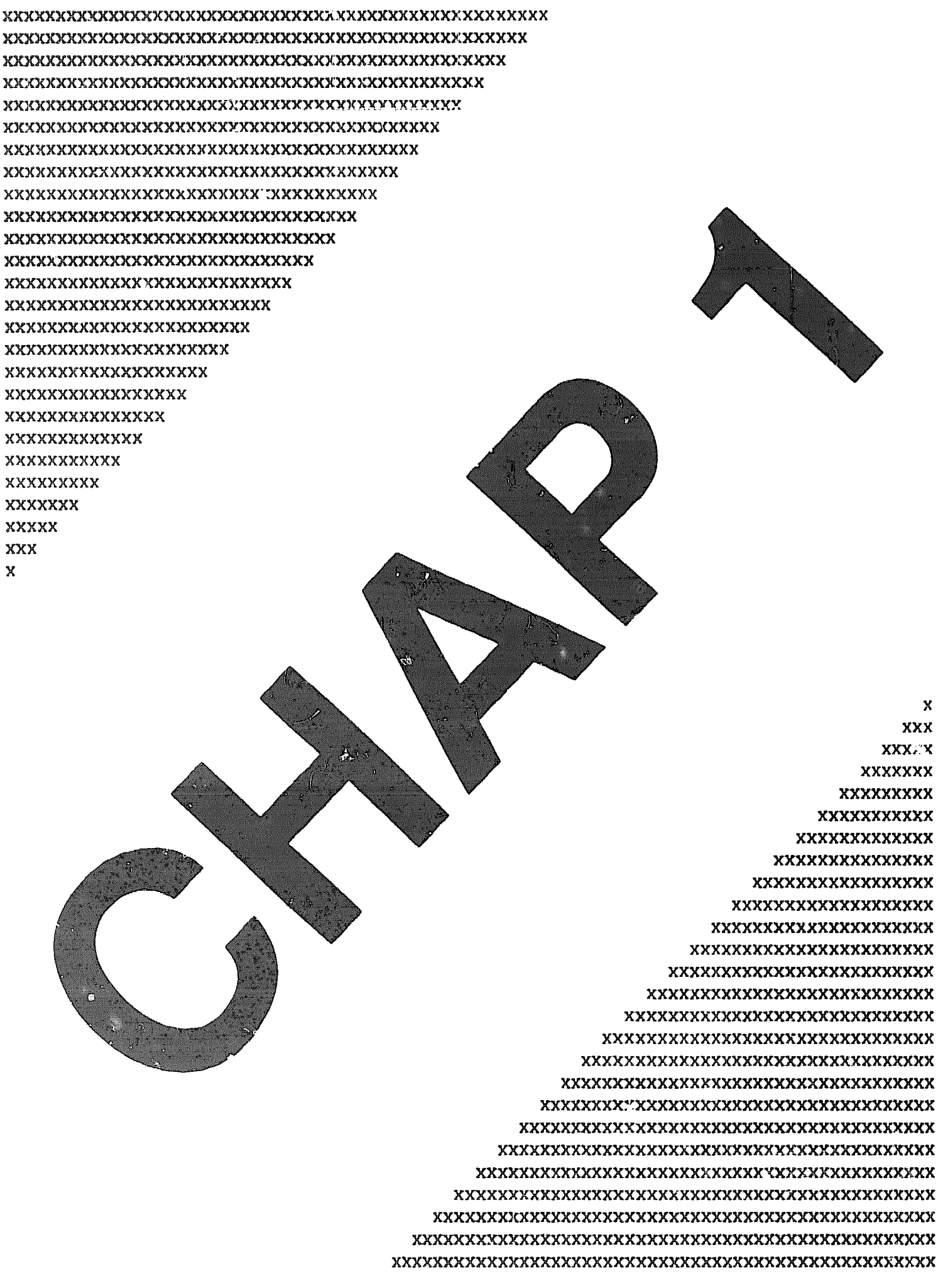
Chapter 6 Provides lists of router hardware and cable options and the physical and electrical specifications.

DECrouter 200 Documents

- *DECrouter 200 Hardware Installation/Owner's Guide*
- *DECrouter 200 Software Installation Guide (VMS/MicroVMS)*
- *DECrouter 200 Software Installation Guide (ULTRIX-32/32m)*
- *DECrouter 200 Management Guide*
- *DECrouter 200 Identification Card*

Associated Documents

- *The Ethernet-A Local Area Network-Data Link Layer and Physical Layer Specifications*
- *Ethernet Installation Guide*
- *Ethernet Networks: Ethernet Products and Services Catalog*
- *H4005 Digital Ethernet Transceiver Installation Manual*
- *H4005 Digital Ethernet Transceiver Technical Manual*
- *DELNI Installation/Owner Manual*
- *Installing Etherjack*
- *Motorola Microprocessor Data Manual*
- *Routing and Networking Overview*


CHAPTER 1

1
DECrouter 200

1.1 DECrouter 200 Introduction

The DECrouter 200 system is an Ethernet-based device that allows network connection for up to eight asynchronous devices. The primary use for the router is to connect IBM personal computer nodes running DECnet-DOS, specifically, the IBM Personal Computer/AT and the IBM PC/XT, to the Ethernet and to wider DECnet networks, using existing office wiring. The router also connects the Digital Rainbow 100 series, and the Digital Professional 300 series of personal computers to the Ethernet and wider area networks, using existing office wiring. The DECrouter 200 also connects any DECnet nodes using asynchronous DDCMP with or without modem control. DECrouter 200 has two interfacing capabilities: from devices or dial-in modems to the Ethernet; or from computer systems or dial-out modems to the Ethernet. Figure 1–1 shows the name panel of the router.

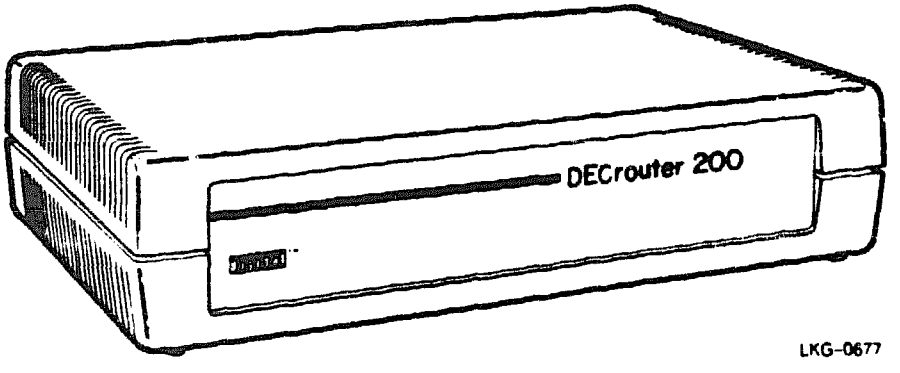


Figure 1–1: DECrouter 200 Name Panel View

1.1.1 Firmware

Two kilobytes of EEPROM store all terminal default parameters and some system default parameters for each communication line. The EEPROM has a limited life of 10,000 write cycles per byte and must be protected from indiscriminate write cycles. Write cycles, which are signaled by an enable bit to the EEPROM, permit a single write cycle, and are only read by the router ROM code. For a detailed description of EEPROM and its functions, see Sections 5.4.11 and 5.4.12.

1.1.2 Programmable Timers

Four 2681 communications dual asynchronous receiver transmitters (DUARTs) provide four 16-bit timers (DUARTs 0 – 3) with the following functions:

- DUART0 – Processor interrupt timer for dynamic memory refresh
- DUART1 – General purpose timer/set to 25 milliseconds
- DUART2 – Watchdog timer for fatal (hard) errors
- DUART3 – Timer for LED status indicators

For a detailed description of programmable timers, see Sections 5.5.13 through 5.5.18.

1.2 Ethernet Interface System

The hardware interface between the DECrouter 200 and the Ethernet is controlled through a chip set consisting of a Local Area Network Controller for Ethernet (LANCE) and a Serial Interface Adapter (SIA). The chip set converts Manchester encoded serial data at 10 Mbits per second into parallel, 16-bit data for memory storage.

1.2.1 Local Area Network Controller for Ethernet (LANCE)

The LANCE is a master/slave, direct memory access (DMA) device that converts 10 MHz of serial data into 16-bit words of parallel data. The LANCE, in slave mode, allows direct memory access from the 68000 data bus. For a detailed description of the LANCE, see Section 5.6.2.

1.2.2 Serial Interface Adapter (SIA)

The SIA is a Manchester encoder/decoder with IEEE and Ethernet specifications. The SIA links the LANCE to the Ethernet transceiver cable and acquires clock signals and data from the Manchester output at 10 MHz. For a detailed description of the SIA, see Section 5.6.1.

1.2.3 Service Node and Load Host

Processors supporting the DECrouter 200 on an Ethernet system provide service nodes that:

- Implement DECnet Phase IV software
- Serve as load hosts to down-line load the DECrouter software image to any requesting router

1.2.4 Firmware and Software

Following is a list of the Programmable Read Only Memory (PROM) code on the DECrouter 200 printed circuit board:

- Self-test program
- Initialize program — down-line loading and up-line dumping
- On-line debugging tool (ODT)

The following down-line load procedure explains how the router acquires the router software image from a load host. The software image is stored in program random access memory (RAM).

- Upon power-up, the Self-test program performs a series of diagnostics and signals for the initialize program.
- The initialize program signals a request for a down-line load, based on the router physical address.
- The DECrouter then loads from the first properly configured Phase IV DECnet system that responds to the request.
- When loaded, the router software operates until the next power-up sequence. When the router parameters are properly set, data can be routed between nodes on the Ethernet and nodes connected to asynchronous ports.

1.3 Connector Panel Element Description

1.3.1 Controls and Connectors

Figure 1–2 is a view of the router showing cable connectors, switches, and LEDs. The information in Table 1–1 explains the function of each LED.

Table 1–1: Connector Panel Elements

The four LEDs on the back of the router function as follows:

	State	Condition
Power — D1	ON	DC voltage OK 5V power
	OFF	DC voltage problem 5V power
Diagnostic — D2	ON	Selftest OK
	OFF	Fatal error or test in progress
	BLINKING	Non-fatal (soft) error
Software — D3	ON	Operating software loaded, no error
	OFF	Down-line load in progress
	BLINKING	Loading failed, will-try-again
Ethernet Carrier — D4	ON	Network active
	OFF	Network not active

NOTE

Soft errors do not hinder minimum operation of the system. Hard errors hinder overall operation of the system. The software LED is only valid when the diagnostics are done.

1.3.2 Switch S1

The switch is not functional on the router.

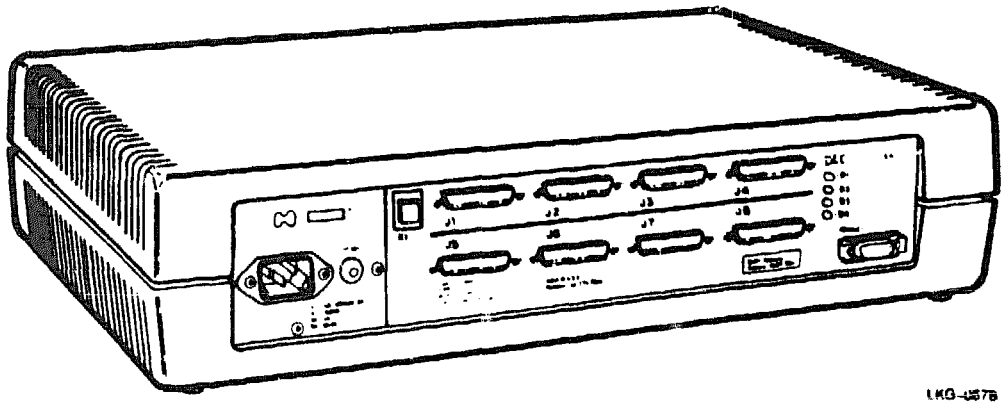


Figure 1–2: DECrouter 200 Cable Connection Panel

1.3.3 Asynchronous Ports (J1 – J8)

Through the eight 25-pin male connectors, the router can accommodate up to eight personal computers or any computer systems that have asynchronous DDCMP protocol. Chapter 6 contains a complete list of standards and protocols for interfacing serial lines and the Ethernet.

1.3.4 Ethernet Transceiver Port

The Ethernet transceiver connector, a 15-pin female type, is on the bottom right corner of the rear panel. The device connects the router to a Digital Ethernet transceiver or a DELNI transceiver cable.

1.3.5 AC Line Voltage Input Selector Switch

The line voltage input selector switch is to the right of the AC power receptacle. The switch allows the installer to set the router for 47–63 Hz input speed as follows:

- 120 Vac, 90 Vrms to 128 Vrms, single phase, 3-wire
- 240 Vac, 176 Vrms to 268 Vrms, single phase, 3-wire

1.3.6 AC Power Cord Receptacle

A country-specific power cord is supplied with the router.

1.3.7 AC Circuit Breaker

The server has a resettable circuit breaker to the right of the power receptacle on the rear of the server.

1.3.8 Power-Up and Initialization

After installation and power application, a terminal can be plugged into J1 to check for Self-test error messages or for down-line load messages. The terminal must be set to the following default characteristics:

- Transmit/Receive = 9600
- Character size = 8 bits
- Parity = None

1.3.9 Initialization Sequence

Application of an AC power source initiates the following power-up sequence:

1. The power LED lights.
2. Self-test begins, and if no soft errors are detected, the Self-test LED lights.
3. The down-line code is invoked and the router software image is down-line loaded. After successful down-line loading, LED2 lights.
4. The router image begins execution.
5. The unit is functioning.

1.4 Ethernet Systems

The Ethernet, as a local area network (LAN), is a communication method based on joint specifications from Digital, Intel Corporation, and Xerox Corporation. For extensive hardware descriptions, refer to *The Ethernet – A Local Area Network – Data Link Layer and Physical Layer Specification*.

The LAN communication system supports a 10-MHz data rate over interconnected coaxial cables. The Ethernet specification defines the physical properties of the coaxial cable, the transceiver, and some of the peripheral hardware. The specification also defines basic rules for network access, device addressing, and data frame format.

1.4.1 System Elements

Figure 1–3 shows the DECrouter used in a large office and computer room environment. DECrouter 200 RTRDEV in Figure 1–3 offers:

- Connection to personal computers
- Connection using existing twisted-pair wiring
- Modems in office area accessing remote personal computers

DECrouter 200 RTR2 in Figure 1–3 offers:

- Connection to large DECnet systems
- Modem connections to remote facilities
- Modems in a computer room for access to the network by remote nodes

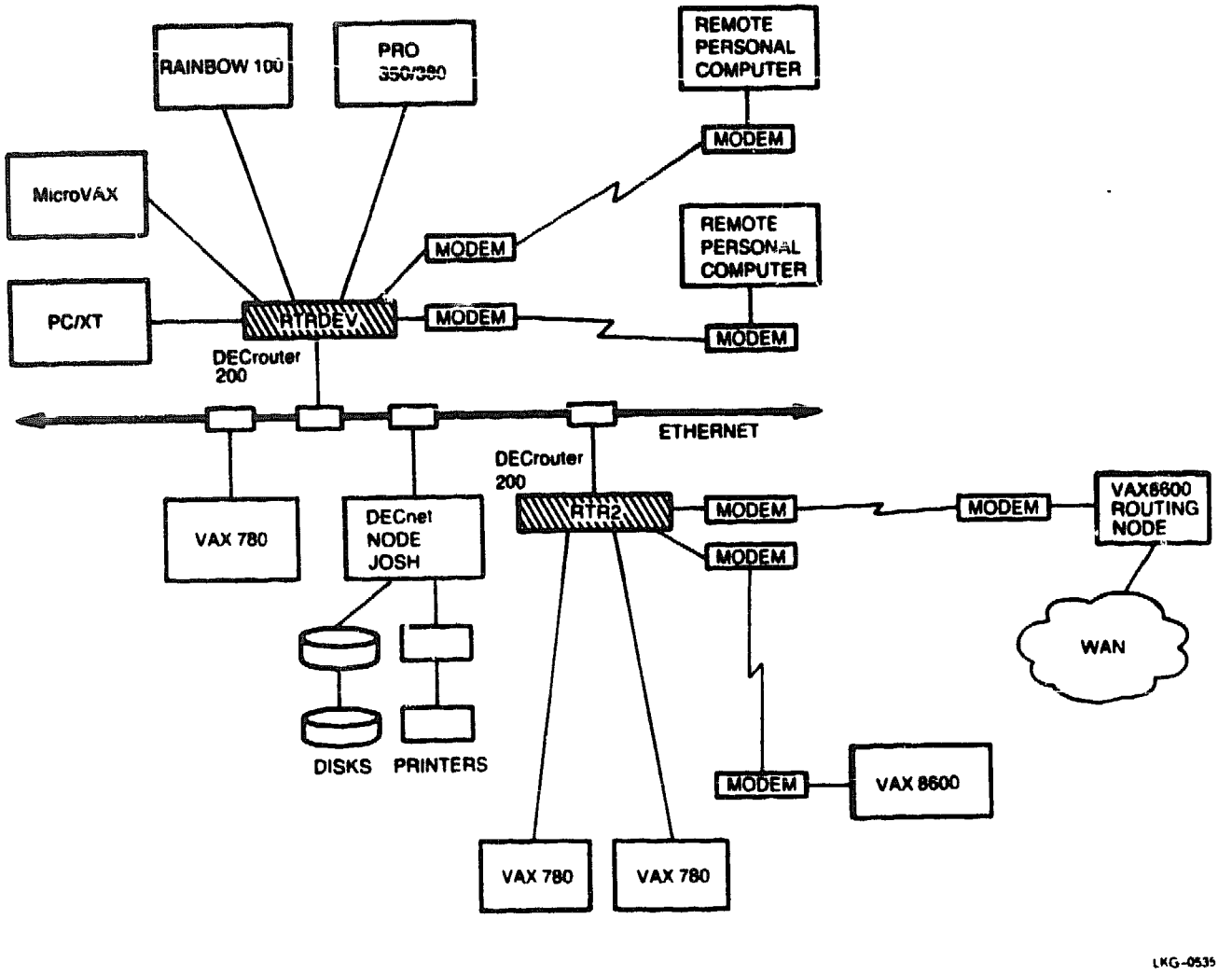


Figure 1–3: DECrouter in a Large Office/Computer Room Environment

Figure 1–4 shows a DECrouter 200 used in a small office environment to connect a few workstations to an office computer. The configuration shows personal computers used as workstations connected to a network that uses a DELNI as the Ethernet.

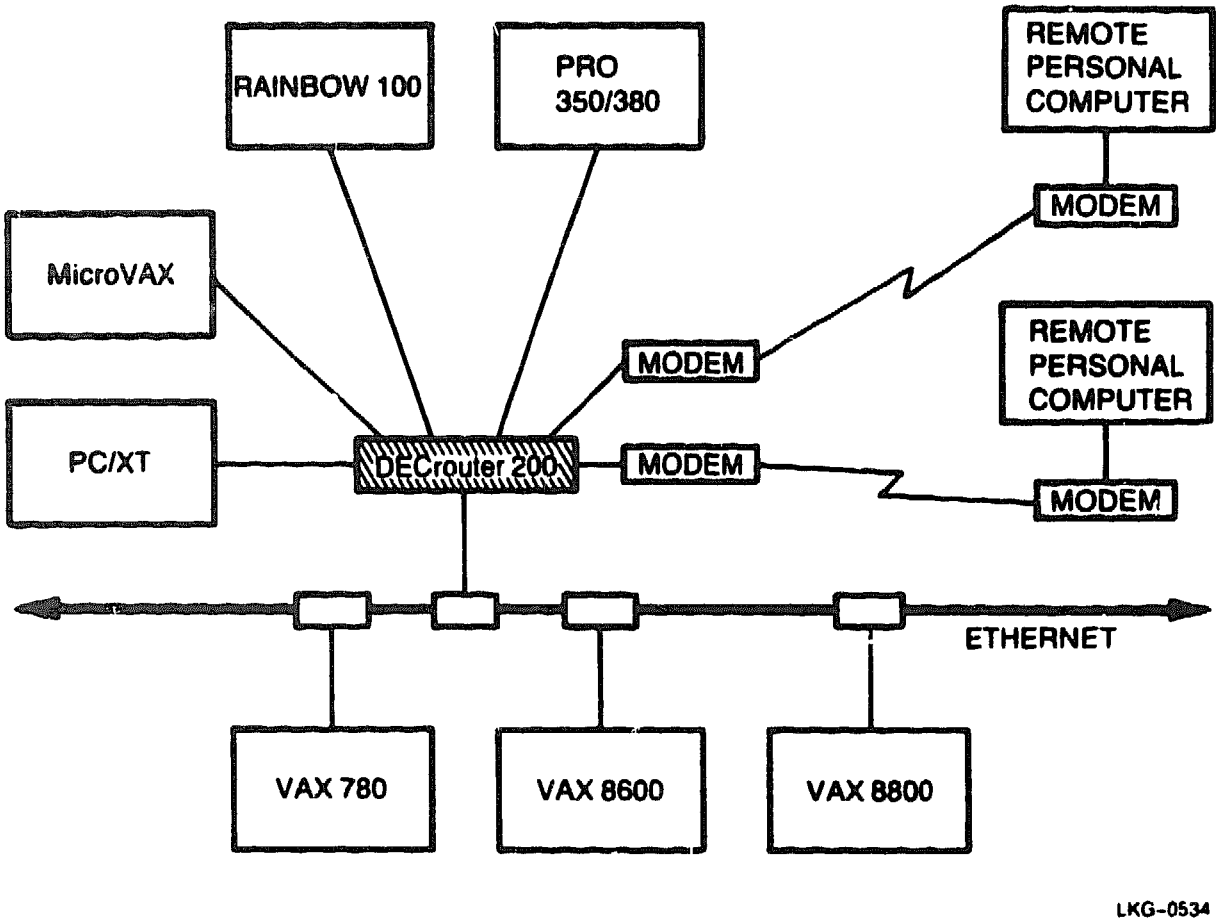


Figure 1–4: The DECrouter in a Small Office/Computer Room Environment

1.4.2 Configuration Criteria

Following are the configuration criteria for optimum network performance on an Ethernet:

- Each coaxial cable segment can be any length up to 500 meters (1640 feet) and must be terminated at both ends at 50 ohms impedance.
- A maximum of 100 stations can be installed on a single cable length. Transceivers must be at least 2.5 meters (8.2 feet) apart.
- Transceiver cables can be any length up to 50 meters (164 feet).
- A repeater can extend the system by connecting two coaxial cable segments through two transceivers.
- The network can be extended using two remote repeaters with an optical point-to-point link of up to 1,000 meters (3,281 feet) as shown in Figure 1–5.

- A maximum of two repeater or remote repeater combinations can be used between any two points on the system.
- A maximum of 1,024 stations can be connected to an extended network. Such a network must be configured with no more than 2.80 km (1.74 miles) separating any two stations on the network, including point-to-point links and transceiver cables.

1.4.3 Communications Methods

Ethernet uses a branching bus topology with a coaxial cable as the transmission medium. Each station transmits a preamble at its start. Receiving stations synchronize and acquire timing from the preamble. The preamble information reduces problems with line skew or variations in carrier frequency. During transmission, a station calculates a cyclic redundancy check (CRC) value, which is appended to the end of the transmission.

1.4.4 Network Access and Communication Protocol

The router supplies power to the Ethernet transceiver on one twisted pair of transceiver cable. Communications between the router and the transceiver are on the following signal pairs:

- Transmit pair — The router encodes and sends Manchester serial data on the transmit pair.
- Receive pair — The router receives Manchester-encoded serial data on the receive pair.

NOTE

The router, operating in half-duplex mode, must be in either transmit mode or receive mode at a given time. When the router is receiving data, the transceiver monitors the coaxial cable for signals and reports the start of a transmission.

- Collision pair — The transceiver reports a collision condition with another station by sending a 10-MHz carrier to the router on the collision pair. The transceiver also tests for router collision detection circuits by sending 10-MHz frequency for approximately 2 microseconds after every transmission. This is called a heartbeat check.

1.4.5 Network Access

There is no priority arbitration for access to an Ethernet. All systems and nodes have equal access Access is gained using a line contention/collision detection protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD).

1.4.6 Collision Detection

Two or more stations transmitting at once nets a collision, which destroys data. When a collision is detected, the transmitting stations continue transmission for a predetermined time. The continuation of transmission is detected by all stations on the network.

Following are frequently used collision detection terms:

- Deference — When a collision is detected, all transmitting stations, and/or stations waiting to transmit, wait for an integral number of slot times, which are determined by an exponential binary backoff algorithm, before trying to transmit again. Each subsequent collision and backoff reduces the possibility of further collisions. This way all stations gain access at random intervals and can complete transmissions.
- Slot Time — The maximum time, 51.2 microseconds, in which collisions are apt to occur on a network. Slot time represents the maximum time from when a station starts to transmit to the time that it detects a collision with another transmitting station on the network. Any collision detected beyond 51.2 microseconds is flagged as a late collision error.
- Collision Circuitry Test (Heartbeat) — For the transceiver: Tests the router collision detection circuitry by sending a 10-MHz frequency load for approximately 2 microseconds after a normal transmission. A collision circuit failure is recorded as a nonfatal hardware error.

1.4.7 Data Frame Format

Each station transmits a frame of serial data in the format shown in Figure 1–5. Each data-byte is right-shifted within the format.

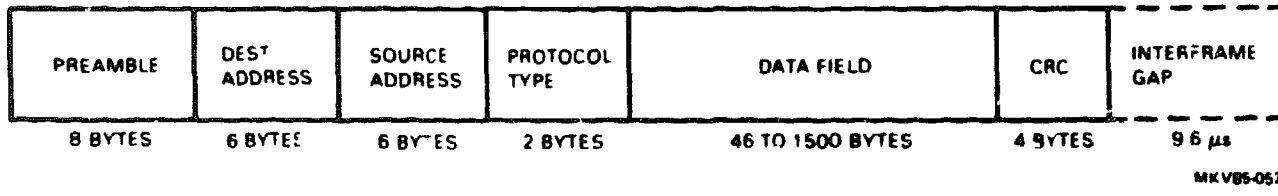


Figure 1–5: Ethernet Data Frame Format

Table 1–2: Ethernet Characteristics

Characteristic	Value/Definition
Topology	Branching bus
Transmission Medium	Coaxial cable using Manchester-encoded digital baseband signaling
Data Rate	10 million bits per second (10 MHz)
Maximum Separation of Stations	2.80 km (1.74 miles)
Maximum Number of Stations	1,024 stations
Network Control	Multi-access with equal distribution to all stations
Access Control	Carrier sense multiple access with collision detection (CSMA/CD)
Data Frame Format	72 to 1526 bytes, including a preamble, with a variable data field of 46 to 1500 bytes

The following list identifies and describes each of the seven Ethernet data frame fields.

1. **Preamble** — A 64-bit (8-byte) pattern of alternating 1s and 0s that allows the receiver circuits to settle. The last two bits are transmitted as 1s for synchronization, indicating that all following information is interpreted as data.
2. **Destination Address** — A 48-bit (6-byte) address that specifies the destination station. Each station has a unique Ethernet address and examines the field to determine if it should accept or reject the data frame.

   If the high order bit of the first byte is a 0, the destination address is unique to each particular station. If the high order bit is a 1, a logical group of recipients or devices can be specified in a multicast address. A broadcast address of all 1s indicates that all stations on the network are to accept the data frame.
3. **Source Address** — a 48-bit (6-byte) address that identifies the transmitting station.
4. **Protocol Type** — A 16-bit (2-byte) code that identifies the client layer protocol associated with the frame when multiple higher level protocols are sharing the network. Protocol type is not interpreted at the Data Link layer.
5. **Data Field** — The data field can have from 46 to 1500 bytes. The 46-byte minimum ensures that all recipients are able to recognize valid data frames and discard smaller sequences as collision fragments.
6. **Cyclic Redundancy Check (CRC)** — A CRC-16 calculation performed on all fields except the preamble and appended to the end of the frame.
7. **Interframe Gap** — Every station must supply an interframe gap of at least 9.6 microseconds on a multiple-frame transmission.

1.4.8 Ethernet Characteristics

The information in Table 1–2 explains the major characteristics of an Ethernet system.


CHAP 2

2
Self-Test

2.1 General

This chapter describes the Self-test diagnostics residing in PROM. Self-test can operate in one of four modes:

- Normal Mode
- Manufacturing Mode
- Initialized Mode
- Fatal Error Mode

Normal Mode, Manufacturing Mode, and Initialized Mode test router hardware. Fatal Error Mode reports fatal router errors.

2.2 Self-Test Operating Modes

2.2.1 Normal Mode

The diagnostic mode under which Self-test typically runs, for example at a customer site, is called Normal Mode operation.

Self-test starts when power is applied to the router. When no fatal errors are detected, Self-test signals, lighting D2, and transfers control to the router firmware. In Normal Mode, Self-test executes in less than 15 seconds.

The primary indication for a fatal error is that D2 does not light. When a fatal error is detected, Self-test enters Fatal Error Mode.

2.2.2 Manufacturing Mode

Self-test checks whether or not the Manufacturing Mode jumper is connected. If the jumper is connected, the following changes occur in the test modes:

- The Self-test program loops continually.

NOTE

D2 lights after the first successful pass of Self-test, and remains on as the program loops, until a fatal error is detected.

- Dynamic memory is more extensively tested.
- The Ethernet port and all terminal ports are tested in the external loopback mode.
- Due to a limited life of approximately 10 K writes per byte, the read/write tests of EEPROM are disabled after the first pass.
- All errors are fatal, causing Self-test to turn off D2 and halt execution.
- Self-test executes in less than 35 seconds.

Manufacturing mode is terminated by removing the jumper and recycling power to the router.

2.2.3 Initialized Mode

Initialized Mode is entered from the router software when Self-test is called by the INITIALIZE command in Privileged Local Mode. Initialization control information is saved, and the Self-test parameters are used or changed as needed. Self-test is accessible from the Load/Dump microcode after an unsuccessful load.

In entry mode, the upper 256 bytes of RAM contain the router data called Initialized-RAM. Initialized Mode starts at the first test module and operates under the parameters specified in the Initialized Mode Parameter Byte. Self-test execution time depends on the parameters specified in this byte. Table 2–1 describes the information in the Initialized Mode Parameter Byte.

Table 2–1: Initialized Mode Parameter Byte

Bits	Mode Name	Test Description
<07>	Long	When set, enables the extended RAM test
<02:00>	LOOP MODE *n*	Select one or a combination of the following test modes:
	Bit	**Test Mode Selection**
	<00>	Loop Self-test — Enables the self-test to loop on itself for a specified number of passes
	<01>	External Loopback Terminal — Enables external loopback tests of the terminal ports
	<02>	Disable NI External Loopback — Disables external loopback testing of the LANCE and SIA chips

2.2.4 Fatal Error Mode

Self-test enters Fatal Error Mode when a fatal error is detected. All errors are fatal in Manufacturing Mode.

The error routine writes the test in error to a byte location in EEPROM and is cleared by the power-up reset procedure.

2.2.5 Error Types

Errors detected by Self-test are classified as either nonfatal or fatal:

- Nonfatal errors are failures that interfere with normal operation or cause the router to operate at decreased capacity. At completion, Self-test causes D2 to blink if a nonfatal error is detected. The error status parameter is written to Initialized-RAM, and identifies the error for evaluation by the router software.
- Fatal errors can disable the router or cause erratic operation. For fatal errors, D2 remains off and error mode is entered. Unexpected traps and all Manufacturing Mode failures are fatal or "hard" errors.

2.2.6 Nonfatal Errors

The following list explains nonfatal errors:

- **EEPROM Checksum Error** — EEPROM is divided into several functional areas with a parameter checksum maintained in each area. Bad data in any or all of these areas are considered nonfatal.
- **LANCE Error (heartbeat or external loopback error)** — When Self-test detects an error while testing the LANCE in external loopback mode, it flags the nonfatal error in the status longword. The external loopback test is disabled by selecting the internal loopback mode in the Initialized Mode Parameter Byte.
- **Asynchronous Port Errors (on usable ports)** — Self-test flags port errors to the router firmware in the status longword. Router software disables ports after a down-line load. If errors are detected on all eight ports, the router is not usable. This is a fatal error condition.
- **Modem Control Signal Error** — Indicates a modem control signal error on one or more ports.

2.2.7 Nonfatal Error Status

Two nonfatal status longwords are written to Initialized-RAM, in the upper 256 bytes of RAM, by Self-test. The first longword contains the EEPROM checksum error status.

The router has 2 K-bytes of EEPROM. There is a checksum field in each of the 22 sections of EEPROM. Sections are divided as follows:

- Router parameters that contain 173 bytes in 1 section
- Diagnostics that contain 20 bytes in 1 section
- Hardware revisions that contain 3 bytes in 1 section

- Port parameters that contain 1120 bytes in 8 sections
- Service parameters that contain 700 bytes in 10 sections
- Encryption parameters in 1 section

In normal mode, checksum errors are nonfatal. A checksum error is indicated when any one of the following bits is set:

- Bit <00> Checksum error in the router parameter section
- Bit <01> Checksum error in the diagnostic section
- Bit <02> Checksum error in the hardware revision section
- Bit <10:03> Checksum error in the port parameter sections
- Bit <20:11> Checksum error in the service parameter sections
- Bit <21> Checksum error in the encryption parameter section

NOTE

When no fatal errors are detected in Self-test, the two longwords are returned to the load/dump microcode. The following table gives the mnemonic and the function of each bit in the second nonfatal error status longword.

Table 2–2: Nonfatal Error Longword

Bit	Mnemonic	Function
<18>	RTF	Reset to Factory is a status bit; when set, it indicates that the contents of EEPROM was reset to the factory default parameters.
<17>	NI	Network Interconnect indicates a heartbeat error was detected while operating the LANCE.
<16>	XLB	This indicates that the External Loopback Test has failed.
<15:08>	MCS	Modem Control Self-test indicates that the modem control hardware on a channel has failed Self-test. In the longword there is a bit for each channel, which is set if the channel fails. MCS errors are always nonfatal because the router always operates when the data leads on the channel still function.
<07:00>	CHAN	Channel Failure; when bit 2 in the longword is set, channel 2 failed Self-test. CHAN errors are nonfatal if at least one channel functions. Failure of all eight channels to pass Self-test indicates a fatal error.

2.2.8 Fatal (Hard) Errors

All errors that are not considered nonfatal, are fatal. Following are some typical fatal errors:

- **Program RAM Data Error** — Any program RAM data error detected by any memory test
- **Program ROM Cyclic Redundancy Check (CRC) Error** — An error detected on the CRC calculation of the PROM
- **Timer Error** — Any failure detected by the refresh or watchdog timer tests
- **LANCE Error** — Any error detected during initialization or on an internal loopback operating test
- **Terminal Port Error** — When no terminal ports are usable, leaving the router inoperable

2.2.9 Using the Diagnostic LED (D2)

Self-test indicates its status using D2. After executing, Self-test leaves D2 in one of three states: ON, OFF, or BLINKING.

- D2 ON – Self-test has completed without detecting any hardware errors.
- D2 OFF – A fatal hardware error was detected.
- D2 BLINKING – A nonfatal hardware error was detected; an error message displays on the console terminal. See the *DECrouter 200 Management Guide* for a complete description of the diagnostic error messages.

In Normal Mode, Self-test executes in under 15 seconds. In Manufacturing Mode, it takes at least 35 seconds. In Initialized Mode, the execution time varies according to the parameters selected in the Initialized Mode Parameter Byte.

2.2.10 Self-Test Program Tests

There is one loop that calls each test, in order. After a test is run, the test returns to the loop. Tests are executed from both RAM and ROM.

Following is the dispatch table, which shows each test by number. The table also shows how each test is invoked, where the test executes, and the test name:

Table 2–3: Self-Test Dispatch Table

Test	Invoked by	Execution	Name
01	Jump	ROM	ST$CPU REG J
02	Jump	ROM	ST$JAM TEST J
03	Jump	ROM	ST$QUICK RAM J
04	Jump	ROM	ST$EXTENDED RAM J
05	Jump	ROM	ST$STUCK INTERRUPTS J

(continued on next page)

Table 2–3 (cont.): Self-Test Dispatch Table

Test	Invoked by	Execution	Name
06	Call	ROM	ST$REFRESH TIMER
07	Call	ROM	LAN$REG TEST
08	Jump	ROM	ST$WD TIMER
09	Call	ROM	ST$GP TIMER
10	Call	ROM	ST$LOADER
11	Call	ROM	ST$PROM CRC
12	Call	ROM	ST$NI ADDRESS
13	Call	ROM	ST$PARITY
14	Call	ROM	ST$EEPROM RW
15	Call	ROM	ST$RESET TO FACTORY
16	Call	ROM	ST$EEPROM CS
17	Call	ROM	ST$MODEM CS
18	Call	ROM	ST$RTS CTS
21	Call	RAM	LAN$REJECT
22	Call	RAM	LAN$ACCEPT PHY
23	Call	RAM	LAN$COLLISION
24	Call	RAM	LAN$RCV GOOD CRC
25	Call	RAM	LAN$XMT CRC
26	Call	RAM	LAN$RCV BAD CRC
27	Call	RAM	LAN$BROADCAST
28	Call	RAM	LAN$MULTICAST
29	Call	RAM	LAN$EXTERNAL LOOP
30	Call	RAM	RT$CHAR LENGTH
31	Call	RAM	RT$BREAK
32	Call	RAM	RT$FRAMING
33	Call	RAM	RT$OVERRUN
34	Call	RAM	RT$BAUD RATE
38	Call	RAM	ST$EXERCISER

2.2.10.1 Processor Register Test — ST$CPU_REG_J — Verifies that there are no *stuck-at* faults in the CPU registers.

2.2.10.2 Self-Test UN-JAM Test — ST$JAM_TEST_J — Verifies that the reset state is cleared. When the router is powered up, the hardware reset signal causes PROM to map to a RAM address space where the CPU fetches its reset vectors and starts execution. Clearing the reset allows the mapping of PROM to its correct address.

2.2.10.3 RAM Quick Verify Test — ST$QUICK_RAM_J — Verifies that each RAM location is addressable and that no locations have stuck-at faults. The algorithm also yields full coverage of shorted or open address lines.

2.2.10.4 Extended RAM Test — ST$EXTENDED_RAM_J — Provides extensive Program RAM tests; the test executes only when in Manufacturing Mode, or when called from LAT software with extensive memory testing enabled. This test identifies stuck-at faults and coupling faults, and ensures that all calls exist.

2.2.10.5 Stuck-at Interrupt Test — ST$STUCK_INTERRUPTS_J — Verifies that no processor interrupts are pending. Processor interrupts can be caused by malfunctioning interrupt logic.

2.2.10.6 Refresh Timer Test — ST$REFRESH_TIMER — Verifies that the refresh timer (DUART0) interrupts at the correct Interrupt Priority Level (IPL) and the correct vector, and that the timer is running in free running mode.

2.2.10.7 LANCE Register Test — LAN$REG_TEST — Verifies that the CPU correctly accesses the LANCE registers, and ensures that there are no stuck-at faults in the LANCE registers, and that the LANCE is correctly reset.

2.2.10.8 Watchdog Timer Test — ST$WD_TIMER — Ensures that the Watchdog Timer resets the module. This function of the Watchdog Timer is tested only on the first pass of Self-test; subsequent passes check whether or not the watchdog timer expires. Resetting of the module is not allowed. Before exiting, the watchdog subsystem is initialized, so that when any portion of Self-test hangs, the timer expires, causing a fatal error.

2.2.10.9 General Purpose Timer Test — ST$GP_TIMER — Verifies that the general purpose timer (DUART1) interrupts at the correct IPL and that the timer and the correct vector, and that the timer is in free running mode.

2.2.10.10 Test Loader — ST$LOADER — To simulate normal router operation, several tests execute in RAM rather than in PROM. The ST$LOADER test copies these tests to RAM and verifies that they were correctly copied.

2.2.10.11 PROM CRC Test — ST$PROM_CRC — Performs a Cyclic Redundancy Check (CRC) on the entire contents of the PROM; compares PROM contents with the value CRC stored in the last longword of the PROM.

2.2.10.12 NI Address PROM Checksum Test — ST$NI_ADDRESS — Performs a checksum calculation on the 32-byte NI Address PROM that holds the router Ethernet address and compares the result with the checksum stored in the PROM.

2.2.10.13 Parity Logic Test — ST$PARITY — Ensures that the parity logic correctly detects and reports various parity errors.

2.2.10.14 EEPROM Read/Write Test — ST$EEPROM_RW — Verifies the read and write functions of EEPROM and tests the write-enable and write-disable logic. EEPROM has a limited life cycle, and writes to the area should be used discriminately. Looping on the portion of the test that writes to EEPROM is not allowed.

2.2.10.15 Reset to Factory Test — ST$RESET_TO_FACTORY — During power-up, when all the LEDs are on, the reset switch (S1) is polled. A flag is set true if the reset switch is pressed during a .5-second interval. This test examines the flag and, if true, the remainder of the test executes.

The contents of EEPROM are restored to its factory default settings. Default parameters in ROM are copied to the EEPROM, and new checksum calculations are performed and included in EEPROM.

2.2.10.16 EEPROM Checksum Test — ST$EEPROM_CS — EEPROM is divided into these sections:

- Router parameters
- LANCE, ECO, and revision parameters
- Diagnostic parameters
- Separate areas for each of the eight terminal ports

This test verifies the accuracy of all but the diagnostic parameter section of EEPROM.

2.2.10.17 Modem Control Signals Test — ST$MODEM_CS — Checks modem control signal logic, ensuring that each bit in every modem control register can be set and cleared; tests each change-detect and interrupt generation circuit for proper functioning.

2.2.10.18 Request-to-Send — Test ST$RTS_CTS — Verifies that the request-to-send and clear-to-send flow control logic functions properly.

2.2.10.19 LANCE Reject Physical Address Test — LAN$REJECT_PHY — Transmits an internal loop packet with a destination address that differs from the LANCE address; checks that the packet is correctly rejected by the LANCE.

2.2.10.20 LANCE Accept Physical Address Test — LAN$ACCEPT_PHY — Transmits an internal loop packet with a destination address equal to the LANCE address; verifies that the packet is correctly transmitted and received.

2.2.10.21 LANCE Force Collision Test — LAN$COLLISION — Checks the LANCE collision detection logic. By setting the RTRY bit in the transmit descriptor, the test verifies that a collision can be forced and reported. A packet is sent in internal loopback, promiscuous mode, with the collision bit set; this enables the LANCE to detect collisions. Transmission is attempted 16 times before the LANCE signals a retry error and terminates the attempt.

2.2.10.22 Receive CRC Logic Test — LAN$RCV_GOOD_CRC — Ensures that the LANCE receives a correct CRC without flagging CRC errors. Packets are transmitted in promiscuous, internal loopback mode. Transmit CRC generation is disabled, and a calculated CRC is appended to the packet. The receiver checks the CRC against the CRC that it calculated as the packet is received.

2.2.10.23 Transmit CRC Logic Test — LAN$XMT_CRC — Ensures that the LANCE generates and appends a CRC on transmission. A packet transmits in promiscuous, internal loopback mode. The transmitter generates the CRC; the receiver does not check the validity of a received CRC. A CRC is manually calculated, and checked against the CRC that the transmitter appended to the packet.

2.2.10.24 Receive Bad CRC Test — LAN$RCV_BAD_CRC — Ensures that the LANCE detects bad CRCs. Packets are transmitted in promiscuous, internal loopback mode with a bad CRC appended to the packet. The receiver flags CRC errors.

2.2.10.25 LANCE Broadcast Address and Byte Swap Test — LAN$BROADCAST — Ensures that the LANCE accepts an internal loopback packet with a destination address equal to the broadcast address. This test is run in Byte Swap Mode.

2.2.10.26 LANCE Multicast Address Test — LAN$MULTICAST — Ensures that the LANCE accepts or rejects frames based on group logical addresses.

2.2.10.27 External Loopback Test — LAN$EXTERNAL_LOOP — Ensures that the LANCE can correctly transmit and receive external loopback packets. This also tests the Serial Interface Adapter (SIA).

2.2.10.28 Length, Parity, Initialize Sequence — RT$CHAR_LENGTH — Tests each channel for the correct interrupting sequence, character length, and parity generation. The receivers are initialized to interrupt on RXRDY. The receiver identifies the different character lengths and the different parity types of the four characters that are transmitted.

2.2.10.29 DUART Transmit/Receive Break Test — RT$BREAK — Tests the ability of each of the Receive/Transmit channels to detect and accurately report a break character.

2.2.10.30 Force Framing Error Test — RT$FRAMING — Tests the ability of each channel to detect various framing errors and executes only in external loopback mode.

2.2.10.31 Force Overrun Error, FIFO Depth Check Test — RT$OVERRUN — Tests that each DUART channel accurately reports a receive overrun error.

2.2.10.32 Baud Rates Test — RT$BAUD_RATE — Tests that each channel transmits and receives at several different baud rates.

2.2.10.33 System Exerciser Test — ST$EXERCISER — Exercises much of the router system hardware. Data transmits and is received on each terminal port and the network interconnect (NI) port while modem control interrupts are posted. The following devices are active while this test executes:

- **General Purpose (GP) Timer** — Interrupts every 10 milliseconds.
- **Refresh Timer** — Interrupts every 3.8 milliseconds.
- **EEPROM** — Accessed to read data from EEPROM. The write inhibit circuitry is exercised without performing a write.
- **RAM and Parity Detection/Generation** — The DUART and LANCE buffers are in RAM, and the RAM and parity detection/generation logic is used during testing.
- **NI PROM, PROM, and I/O Blocks** — Accessed during testing.
- **Watchdog Timer** — Active during testing.
- **LANCE** — Accessed while in operation.

2.2.11 Self-Test Error Codes

Error information is stored in Initialized-RAM, with the error information start address at $8014A. A Failing Test byte indicates the test in which an error is detected. For example, if ST$QUICKRAM fails, test number 3 is written to the Failing Test byte, as shown in Table 2–2. The error number identifies the part of the test in which an error is detected. A test can generate decimal error codes from 1 to 99.

The ERROR DATA section contains test specific data, for example, when RAM TEST detects an error while writing to location $1040. Error data contains the address of the failures, and the bits in error.

Table 2–4 categorizes and defines the various Self-test error codes.

Table 2–4: Self-Test Error Codes

Code	Message	Description
Pass/Fail Error Codes		
00	*st$k_success*	Operation successful
01	*st$k_failure*	Operation failed
General Error Codes		
1	*err 1*	First error check failed
2	*err 2*	Second error check failed
3	*err 3*	Third error check failed
4	*err 4*	Fourth error check failed
5	*err 5*	Fifth error check failed
6	*err 6*	Sixth error check failed
7	*err 7*	Seventh error check failed
8	*err 8*	Eighth error check failed
9	*err 9*	Ninth error check failed
DUART Error Codes		
11	*rt$k_xmt_timeout*	A transmit interrupt did not occur on time
12	*rt$k_rcv_timeout*	A receive interrupt did not occur on time
13	*rt$k_rcv_char_err*	A bad character was received
14	*rt$k_rcv_stat_err*	A bad status was received
15	*rt$k_cum_failure*	All ports failed
16	*rt$k_icnt_err*	Number of interrupts incorrect

(continued on next page)

Table 2–4: (Cont.) Self-Test Error Codes

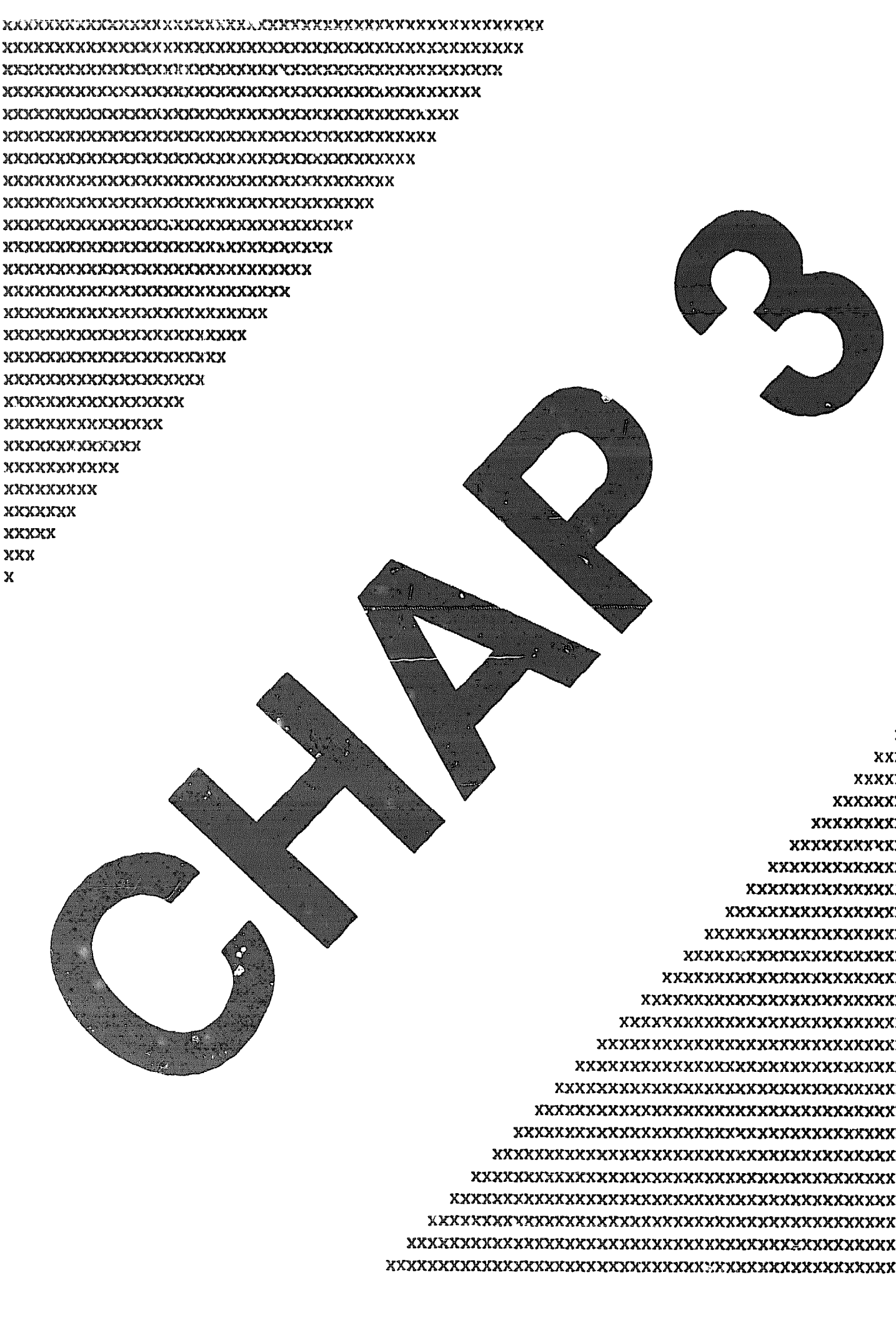
Code	Message	Description
LANCE Error Codes		
21	*lan$k_no_read_init*	The initialize block was not read
22	*lan$k_rcv_desc_err*	The error bit is set in the receive descriptor
23	*lan$k_rcv_desc_timeout*	A timeout occurred for possession of the descriptor
24	*lan$k_rcv_timeout*	A receive interrupt did not occur
25	*lan$_diff_buff_size*	The transmit and receive buffers are different in size
26	*lan$_diff_buffers*	The transmit and receive buffers are different
27	*lan$_bad_rcv_buff_crc*	An incorrect CRC is stored in the receive buffer
28	*lan$_xmt_desc_err*	The error bit is set in the transmit descriptor
29	*lan$_xmt_desc_timeout*	A timeout occurred for possession of the descriptor
2A	*lan$_xmt_timeout*	A transmit interrupt did not occur on time
2B	*lan$_csr0_err*	An error bit is set in CSR0
2C	*lan$_no_stop*	LANCE did not stop
2D	*lan$_no_exp_err*	LANCE did not flag an expected error
Asynchronous Error Codes		
70	*st$k_ui_err*	Unexpected interrupt error
71	*st$k_par_err*	Parity error
72	*st$k_unx_rt_int*	Unexpected DUART interrupt
73	*st$k_warm_err*	Illegal warmstart reset
74	*st$k_warm_err*	Illegal warmstart and a bad stack

NOTE

Typical asynchronous errors are unexpected interrupts and parity errors. Error codes are written to EEPROM under Error Number. Following is an example of an EEPROM error display when Self-test fails:

```
Failing Test:    03 @ 8014B
Error Number:    71 @ 8014D
Error Data:      00 @ 8014F
```

This indicates that Test 3 (ST$QUICK_RAM) failed due to a parity error, 71(hex).


СНАЙПЕР 3

3
Initialize Program

3.1 Introduction

The initialize program is in the Program ROM and starts on completion of the Self-test program or on detection of a fatal bugcheck error.

After Self-test, the initialize program transmits a request for a down-line load of the router image from a load host. When Self-test detects a fatal hardware error or the Ethernet loopback test failed, no request is sent. An active DECnet Phase IV host is needed to operate the router.

The initialize program is started by the router manager and the program issues a request for an up-line dump of Program RAM on a fatal bugcheck. The watchdog timer also initializes up-line dump requests.

The initialize program supports the following router start-up and operating functions:

- Part 1 – transmits a request program load message on the Ethernet for a down-line load of the router software image.
- Part 2 – processes down-line load records from the load host, stores the router software in program RAM, and displays the load status on the terminal.
- Part 3 – displays a fatal bugcheck message on the terminal when a fatal error condition is detected by the operating software.
- Part 4 – when enabled, transmits an up-line dump of all router program RAM locations to a dump host after a fatal bugcheck.

3.2 Down-Line Load

The initialize program displays messages on the terminal to show the progress of a down-line load procedure.

3.2.1 Down-Line Load Messages

When Self-test does not detect errors, LED 3 lights and stays on. When a software error occurs, LED 3 blinks signaling that operating software has not been loaded. When the load starts, the router Ethernet address and firmware version number is displayed. A second message gives the status of the image load. The second message repeats every 30 seconds when the load fails, or when a load host volunteer is not available.

The following sequence of messages appears on the terminal screen at 30-second intervals:

```
Local -901- Initializing DECrouter 08-00-2B-00-XX-XX ROM BL7 H/W REV A.A
Local -902- Waiting for image load
```

When a load host volunteer responds, the load procedure is started and the following message, which identifies the load host Ethernet address, is displayed:

```
Local -903- Loading from 08-00-2B-00-XX-XX
```

When the load procedure is interrupted, or the down-line load message is not received within 30 seconds, the following message is displayed, restarting the load procedure:

```
Local -912- Load failure, timeout
```

On completion of the load, the following message is displayed and the program control transfers to the router software, or to ODT, if enabled:

```
Local -904- Image load complete
```

3.2.2 Down-Line Load Procedure

The down-line load procedure runs under the standard Maintenance Operations Protocol (MOP). The process between the router and the load host volunteer follows this sequence:

1. With Self-test complete, and without a hard error or an Ethernet loopback failure, the router transmits a request program load message to the load assistance volunteer multicast address.
2. Host systems that support down-line load search the DECnet database for the router Ethernet address and try to load the file defined in the database entry.
3. The router selects the first assistance volunteer it receives and retransmits the request program load message, addressed to the selected assistance volunteer, which is the load host.
4. The load host initiates the down-line load by sending the router a memory load message with the first record of the router software image file.
5. The router loads the first memory load message and sends a request memory load message to the linked load host.

6. The linked load host sends subsequent memory load messages containing sequential router software records. The router responds to each memory load message with a request memory load message.

7. When all records are loaded in the router, the load host sends the parameter load with a transfer address message and terminates its down-line load process.

8. The router responds with a final request memory load message and starts the acquired router software as fully operational.

3.2.3 Up-Line Dump

The router manager enables the up-line dump parameter. The initialize program sends an up-line dump request to the original load host on a fatal bugcheck. The watchdog timer can also issue an up-line dump request.

During up-line dumping, the initialize program transmits an image of the entire router memory contents, including the router software, using MOP. The procedure also displays a series of messages on the terminal that report the progress of the dump.

3.2.4 Up-Line Dump Messages

Following is the message displayed when the router requests an up-line dump:

```
Local -905- Waiting for image dump
```

When the dump host responds and the up-line dump is started, the following message is displayed to identify the dump host Ethernet address:

```
Local -906- Dumping to host AA-00-00-00-00-00
```

On completion of a dump, the following message is displayed and the program control transfers to the Self-test program, or to ODT if enabled:

```
Local -907- Image dump complete
```

3.2.5 Up-Line Dump Procedure

The up-line dump process between the router and the dump host is based on the following sequence:

1. To initiate a dump, the initialize program sends a **request dump service** message to the original load host. When the original load host services the request, the host sends a **request memory dump** message to the router and goes to step 5 in the sequence, eliminating steps 2, 3, and 4.

2. If the original load host does not respond to the request dump service message within 15 seconds, the router retransmits the request dump service message to the load assistance volunteer multicast address.

3. Host systems that support the request dump service message first verify if they support dumps from the requesting router. Dump hosts that volunteer send an assistance volunteer message to the router.

4. The router selects the first assistance volunteer that responds and retransmits the request dump service message, which is addressed to the selected assistance volunteer.
5. The linked dump host starts the up-line dump sequence by sending a request memory dump message to the router.
6. The router responds to each request memory dump message from the host with a sequential memory dump record containing the image data of each section of program RAM. All router memory contents are dumped.
7. The linked dump host receives each memory dump message, stores the image in sequential records in a router dump file, and sends another request memory dump message to the router
8. The process continues until the router receives a dump complete message from the dump host, and the dump host terminates its up-line dump process. The router enters ODT, if enabled for postdump analysis, or enters the Self-test program.

3.3 Status and Error Messages

Following down-line load, the router software controls and starts operation by displaying any nonfatal, software errors reported by the Self-test. Certain types of Self-test failures may cause the down-line load procedure to abort or restart.

3.3.1 Ethernet Loopback Error Messages

When an Ethernet loopback test fails under the Self-test, a down-line load is not started, and the following message is displayed:

```
Local -910- Image load not attempted, network communications failure
```

A power-up or router reinitialization (CTRL/P) is needed to clear the condition.

Error message 910 indicates that the router cannot access the Ethernet, maybe due to transceiver or cable problems. Pressing (CTRL/P) reinitializes the router and starts another down-line load request. Reinitialization occurs only when down-line load fails. Pressing (CTRL/P) has no effect if the router software starts successfully after a down-line load. This prevents unauthorized users from reinitializing the router during operation or during a down-line load.

3.3.2 Nonfatal Errors

When Self-test does not detect any errors, LED 3 lights, and the initialize program is flagged.

When any nonfatal errors are detected, LED 3 blinks and the initialize program is flagged. The initialize program displays the following message:

```
Local -911- WARNING -- Nonfatal hardware error detected
```

For port errors the following error messages are displayed:

```
Local -920- Parameter checksum error on port n
Local -921- Factory-set parameters applied to port n
```

3.3.3 Load Failure or Timeout Error Message

When a load procedure is interrupted or the down-line load message is not received from the host within 30 seconds, the following message displays and the load procedure restarts:

```
Local -912- Load failure, timeout
```

3.3.4 Fatal Bugcheck Error Message

A fatal bugcheck displays as:

```
Local -913- Fatal Bugcheck PC=XXX, SP=XXX, SR=XXX, MEM=XXX, CODE=XXX
```

where:

PC=	is the contents of the CPU program counter
SP=	is the contents of the CPU stack pointer
MEM=	is the illegal memory address on an addressing error or the address of the instruction that caused the error
CODE=	is the reason for the failure (see Table 3–1)

Table 3–1: Router Crash Error Codes

Code	Description
CPU Exceptions	
2	Bus error
3	Address error
4	Illegal instruction
5	Divide by zero
6	CHK instruction
7	TRAPV instruction
8	Privilege violation
9	Trace
A	Line 1010 emulator
B	Line 1111 emulator
C	Other
D	Spurious interrupt
Self-Test Bugchecks – Program ROM	
11	NI port hardware memory error
12	NI port initialization timeout error
14	NI port transmit buffer error
15	Stack value incorrect in idle loop
22	Unlink error
24	Unable to allocate XCB
31	Command completion error
32	Local output completion error
Router Software Bugchecks – Program RAM	
200	Router software checksum error
211	NI port hardware memory error
212	NI port initialization
214	NI port transmit buffer error
215	Stack value incorrect in idle loop
216	Unlink error
217	Deallocate error
218	Unable to allocate XCB
219	Command completion error
220	Local output completion error
221	EEPROM write block error
222	Entry on output queue with no slots
223	Transmit too long
224	Cannot find status
225	No available circuit control blocks
226	Low pool allocation error
227	Illegal local output state
228	Service defined with no nodes
229	Duplicate node/service name found

3.3.5 Timeout, Abort Dump Message

When timeout occurs during an up-line dump, the following error message displays and the dump aborts. Program control transfers to the Self-test, or to ODT when enabled.

```
Local -914- Timeout, dump aborted
```

3.3.6 Load or Dump Failure Message

The following error message displays when a down-line load or an up-line dump process fails to transmit a load or a dump message after 10 attempts:

```
Local -915- Transmission failure after 10 attempts
```

For a down-line load, the procedure restarts. For up-line dumping, the process is aborted and program control transfers to the Self-test, or ODT when enabled.

3.3.7 Bad Image File Message

The error message:

```
Local -916- Invalid image file, load aborted
```

is displayed on either of these two conditions:

- A down-line load memory load message specifies an odd memory address or an address in the interrupt vector area.
- A parameter load with a transfer address message specifies an odd transfer address.


CHAP 4

4
On-Line Debugging Tool (ODT)

4.1 On-Line Debugging Tool

The on-line debugging tool (ODT) is a specialized console program that resides in the router firmware (ROM). The user enters ODT only at the console terminal, either from the router operating software or from Self-test manufacturing mode. When enabled, ODT is also accessible after a down-line load or a fatal bugcheck.

ODT provides the local input language for router managers or router personnel and takes the place of most of the keys and switches normally on a computer front panel. ODT is for diagnostics specialists for ongoing reliability test (ORT) procedures. ODT is also for authorized router, system, or network managers in isolating hardware network problems.

4.1.1 Initial Requirements

The following must be met before using ODT.

- When entered from the Self-test manufacturing mode, the default console terminal, port 1, must be configured to the factory-set characteristics stored in EEPROM: 8-bit characters with no parity at 9600 baud.
- From the router software, ODT uses the preset characteristics for the console terminal.

4.1.2 Entering ODT

ODT is started from any of the conditions or procedures in the following sections. When invoked, ODT returns the asterisk (*) prompt.

4.1.2.1 Entering ODT from the Self-Test Manufacturing Mode — In manufacturing, the operator enters ODT by pressing <CTRL/SHIFT/S> on the console keyboard: the CTRL and SHIFT keys are held down while pressing the S key. The operator can then run single-step, or loop on the Self-test program. The operator can also set test parameters and loop on errors.

4.1.2.2 Entering ODT from Router Operating Software — Before entering ODT, the operator issues the following command to the router configuration program:

```
DRCP> ENABLE ODT
```

See the *DECrouter 200 Management Guide* for more information on the configuration program. Note that this guide does not describe the ENABLE ODT command.

While enabled, ODT executes on any of the following conditions:

- A BREAK key is typed at the console
- A system crash
- A router reinitialization – when enabled
- The execution of a previously set ODT breakpoint

4.2 ODT Command Functions

The ODT user can issue ODT commands, one to three characters long, at the asterisk (*) prompt. a carriage return or a line feed terminates the command ODT interprets commands and outputs values in hexadecimal.

4.2.1 Command Input Errors

On a command input error, ODT responds with one of the following error message codes:

- **RE (Range Error)** – This means either the lower memory dump address is higher than the highest specified address or the specified CPU register address is out of range (0 – 7). A range error also indicates the ODT user is trying to examine a nonexistent location in RAM. any address from 60000 to 7FFFF
- **CE (Command Error)** – This indicates that an unrecognized command was issued

4.2.2 Command Summary

All ODT commands are in one of the following groups:

- Memory and device register commands
- CPU register commands
- Dump commands
- Program control commands
- Breakpoint commands

Table 4–1 describes the ODT commands

Table 4–1: ODT Commands

Command	Name	Function
Memory and Device Register Commands		
E *r*	Examine	Opens and displays the contents of a word address where *r* is the address of the word being examined. An odd address is masked to produce an even address.
EB *r*	Examine Byte	Opens and displays the contents of a byte address where *r* is the even or odd address of the byte being examined.
E@ *r*	Examine Indirect	Opens and displays the contents of a longword address where *r* is the address of the longword address being examined. An odd address is masked to produce an even address.
RET	Carriage Return	Terminates an ODT command or closes an open address.
LF	Line Feed	Closes the open address and displays the next sequential address.
ESC	Escape	Causes the current data displayed in the E@ command to be used as an address for the next E@.
CPU Register Commands		
$An or $D*n*	Open Register	Opens and displays the contents of a CPU address or data register, where *n* specifies the register number 0 – 7.
$SR	Open Status Register	Opens and displays the contents of the CPU status register (SR).
Dump Commands		
D *r k*	Memory Dump	Displays the contents of a block of memory addresses. In the display, *r* is the first address of the block and *k* is the last address of the block.
CTRL/C	Halt Dump	Halts a memory dump and returns to the ODT prompt (*).
RD	Register Dump	Displays the contents of the internal CPU registers.
Program Control Commands		
G *r*	Go	Starts or continues program execution after a program or breakpoint halt. Execution starts at location *r*. If *r* is omitted, execution continues from the current PC address.
SS	Single Step Enable	Enables the single step mode which allows program execution one step at a time.
NS	Single Step-Disable	Disables the single-step mode.

(continued on next page)

Table 4–1 (cont.): ODT Commands

Command	Name	Function
Breakpoint Commands		
B*n r*	Breakpoint Set	Sets or changes a breakpoint where *n* is the breakpoint number, 0 – 7, and *r* specifies the memory address for the breakpoint. Breakpoints 0 – 7 allow a software trap from up to 8 locations.
BC*n*	Breakpoint Clear	Clears breakpoint *n*, 0 – 7; if *n* is not specified, all eight breakpoints clear.
$B	Display Breakpoints	Displays the first breakpoint table entry. The other seven entries contain the second through the seventh breakpoint addresses and can be displayed in sequence using the LF key.

4.3 Accessing Router Address Space

The router manager or any privileged operator has access to all terminal router memory and device addresses shown in Table 4–2

Table 4–2: Router Address Ranges

Router Memory or Device	Physical Address Ranges in Hexadecimal
Program RAM	000000 – 07FFFF
EEPROM	080001 – 080FFF
DUART0	100001 – 10001F
DUART1	100021 – 10003F
DUART2	100041 – 10005F
DUART3	100061 – 10007F
MODEM	1000A1 – 1000AB
CONF REG	1000C1
LANCE	100080 – 100082
EPROM	180000 – 187FFF
ADPROM	1C0001 – 1C003F

An ODT command accepts up to 6 hexadecimal digits in the address specifier. Chapter 5 gives the procedures to follow for accessing PROM, Program RAM, DUART, and LANCE internal registers. CPU internal registers are not part of the external memory or device address space; the ODT user accesses internal registers separately through ODT

4.3.1 Memory and Device Register Commands

Only one memory or device register location can be opened at a time to display contents for examination or changes. Opening a register location while another is open closes the first register location.

4.3.1.1 Examine (E) Command — Type E, a space, and the address to open a word location. Typing an odd address masks the low-order bit and opens the even word address. Press the RET key to terminate the command.

Example:

```
*E 1000 RET          - Entry
001000 = 0123        - Response
```

When command location 1000 is open, and no change is required, press RET to close the location.

To change the contents of the open word location, type the new contents before pressing RET to close the location.

Example:

```
*E 1000 RET          - Entry
001000 = 0123 3210 RET   - Response and change
```

Press the Line Feed LF key instead of RET key to close the open word location and to open the sequential word location.

Example:

```
*E 1000 RET          - Entry
001000 = 0123 3210 RET   - Response
001002 = 4567        - Response
```

4.3.1.2 Examine Byte (EB) Command — To open a byte location, type EB, a space, the address, and RET.

Example:

```
*EB 1000 RET         -Entry
001000 = 23          -Response
```

or

```
*EB 1001 RET         -Entry
001001 = 01          -Response
```

If the command location 1000 is open and no change is needed, press RET to close the location. ODT issues the prompt (*) and the user can issue a command.

When the contents of the open byte location are changed, type the new contents before closing the location.

Example:

```
*EB 1000 RET         -Entry
001000 = 23 54 RET   -Response and change
```

Press Line Feed (LF) instead of (RET) to close the open byte location. This opens and displays the next sequential byte location.

Example:

```
*EB 1000 (RET)          -Entry
001000 = 23 (LF)        -Response
001001 = 01             -Response
```

4.3.1.3 Examine Indirect (E@) Command — To examine an address, type E@, space, and the hexadecimal value of the address you need to examine.

NOTE

The Examine Indirect command is terminated with (RET), (ESC), or (LF). Change the contents of a location by using (ESC) or (LF).

The following example shows address examination using ODT:

```
*E@ 1000 (RET)          -Entry
2000 = 100 (ESC)        -Response
100  = 800 (ESC)        -Response
800  = 1000 (ESC)       -Response
1000 = 2000 (LF)        -Response
1004 = 800 (RET)        -Entry
*                       -Response
```

The following example shows address location modification using ODT:

```
*E@ XYXY (RET)          -Entry
1000 = 2000 800 (ESC)   -Response
800  = 1000 2000 (ESC)  -Response
2000 = 100 1004 (LF)    -Response
2004 = 5000 (RET)       -Response
*
```

4.3.2 CPU Register Commands

4.3.2.1 CPU Address Register ($A*n*) Command — The following command format opens and displays the CPU address registers A0 – A7:

```
*$An (RET)              -Where n is an address register 0 – 7
```

Example:

```
*$A5 (RET)              -Entry
A5 = 00001FFF           -Response (contents of address
                         register A5)
```

4.3.2.2 CPU Data Register ($Dn) Command — The following command format opens and displays one of CPU data registers D0 – D7:

*$D*n RET	–Where *n* is a data register 0 – 7

Example:

*D0 RET	–Entry
D0 = 00001234	–Response (contents of data register D0)

When these registers are open, the contents are visible. To change the register contents type a new value before pressing LF or RET as described under the Examine (E) command.

Any value for *n* outside the specified range of 0 – 7, produces a range error (RE) message, followed by the prompt *.

4.3.2.3 CPU Status Register ($SR) Command — Open the CPU status register with the following command format:

*SR RET	–Entry
SR = 2701	–Response

Use the same procedure to change the status register.

4.3.3 Dump Commands

Dump commands, in troubleshooting, display program and hardware status.

4.3.3.1 Memory Dump (D) Command — The D command dumps both the contents of a block of terminal router memory and the contents of DUART address registers. LANCE address space can be included in a dump. LANCE registers are not accessed with the D command and are always read as zeros.

A memory dump is initiated by typing a D, a space, the low address, a space, and the high address, as shown in the following format:

*r k RET	–Where *r* is the address of the first location and *k*, the address of the last location

The specified addresses are then dumped in word format as shown in the following example:

*D 1000 102E RET	–Entry

```
001000  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  aaaaaaaaaaaaaaaa
001010  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  aaaaaaaaaaaaaaaa
001020  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  nnnn  aaaaaaaaaaaaaaaa
```

In the example, *n* represents the hexadecimal value for each nibble (4 bits) of the dumped memory locations. Each *a* represents the ASCII value of each n-pair or byte. A period in any *a* position indicates a nonprinting character.

The word address of the first location dumped on a line is displayed at the start of each line. A maximum of eight sequential words are dumped per line. If an odd word address is issued, the low-order bit is masked to make an even address. If the low address issued is greater than the high address, the Range Error (RE) message is displayed and the prompt for another command is given.

4.3.3.2 Halt Dump (CTRL/C) Command — If an error is made while typing in an address range, press CTRL/C to halt a memory dump and ODT returns to the prompt *.

4.3.3.3 Register Dump (RD) Command — Use the RD command to dump the word (16 bits) contents of the following CPU internal registers:

- Address registers (A0 - A7)
- Data Registers (D0 - D7)
- Program Counter (PC)
- Status Register (SR)

Register contents are displayed in the following format:

*RD RET

```
A0 = nnnnnnnn  A1 = nnnnnnnn  A2 = nnnnnnnn  A3 = nnnnnnnn
A4 = nnnnnnnn  A5 = nnnnnnnn  A6 = nnnnnnnn  A7 = nnnnnnnn
D0 = nnnnnnnn  D1 = nnnnnnnn  D2 = nnnnnnnn  D3 = nnnnnnnn
D4 = nnnnnnnn  D5 = nnnnnnnn  D6 = nnnnnnnn  D7 = nnnnnnnn
PC = nnnnnnnn  SR = nnnn
```

In the example, *n* represents a hexadecimal digit. Register A7 is always the supervisor stack point because the Self-test and LAT programs execute in Supervisor Mode. None of these registers are opened for changes: the CPU register commands must be used to change register contents.

4.4 Programming in ODT

The following several sections describe ODT command functions for debugging under the Self-test program or the LAT operating software.

4.4.1 Program Control Commands

4.4.1.1 Go (G) Command — Start the program or continue its operation after a breakpoint or program halt by issuing the Go command:

*Gr RET — Where *r* is the location to begin the program. If the *r* is omitted, the program begins at the address specified by the current PC. Issuing an odd address as the new PC forces the masking of the low order bit to produce an even address.

4.4.1.2 Single Step Enable (SS) Command — When single step is enabled, enter each instruction in sequence using the following format:

*SS RET — Entry

After each step, ODT displays a message in the following format:

SS*r* — Where *r* is the value of the current PC

The Go (G) command can be used to start or to continue program execution.

Example:

*SS ↵	-Entry – single step command
*G 1000 ↵	-Entry Go starts the program at location 1000
SS 001002	-Response – Go executes on instruction, and displays SS, then the updated PC value
*G ↵	-Entry – Go continues, executing the next command
SS 001004	-Response
*	-Prompt for next command

4.4.1.3 Single Step Disable (NS) Command — The following command disables single step:

*NS ↵	-Entry – ODT leaves single step mode

4.4.2 Breakpoint Commands

The ODT program maintains a breakpoint table that can be displayed but not opened for changes. Defined breakpoint commands must be used to make changes in the breakpoint table.

Eight breakpoints, from 0 to 7, can be set. Breakpoints allow the router software to trap and suspend operation from up to eight locations. A breakpoint can be set in any location acquired from program RAM as an instruction but not as data. When a breakpoint is set, ODT replaces the contents of the breakpoint location with a trap instruction that suspends program operation and returns to ODT.

4.4.2.1 Breakpoint Set (B*n*) Command — Set or change breakpoints by using the following command:

*B*nr*	-Where *n* is the breakpoint number and *r* is the breakpoint location

Example:

*B2 1000 ↵	-Breakpoint entry – program halts operation when fetching at location 1000

4.4.2.2 Breakpoint Clear (BC*n*) Command — Clear breakpoints by using the following command:

*BC*n* ↵	-Where *n* is the number of the breakpoint to be cleared – if the issued value for *n* is not within the range, ODT gives the Range Error >>RE prompt. If *n* is not specified, all breakpoints are cleared.

4.4.2.3 Display Breakpoints ($B) Command — Use the $B command to display the first entry in the breakpoint table. All following breakpoints are then displayed in sequence using Line Feed (LF).

Example:

*$B (RET)	-Entry
BK 1 = 0010F0 (LF)	-Response and (LF) entry
BK 2 = 001200 (RET)	-Response (RET) terminates the command

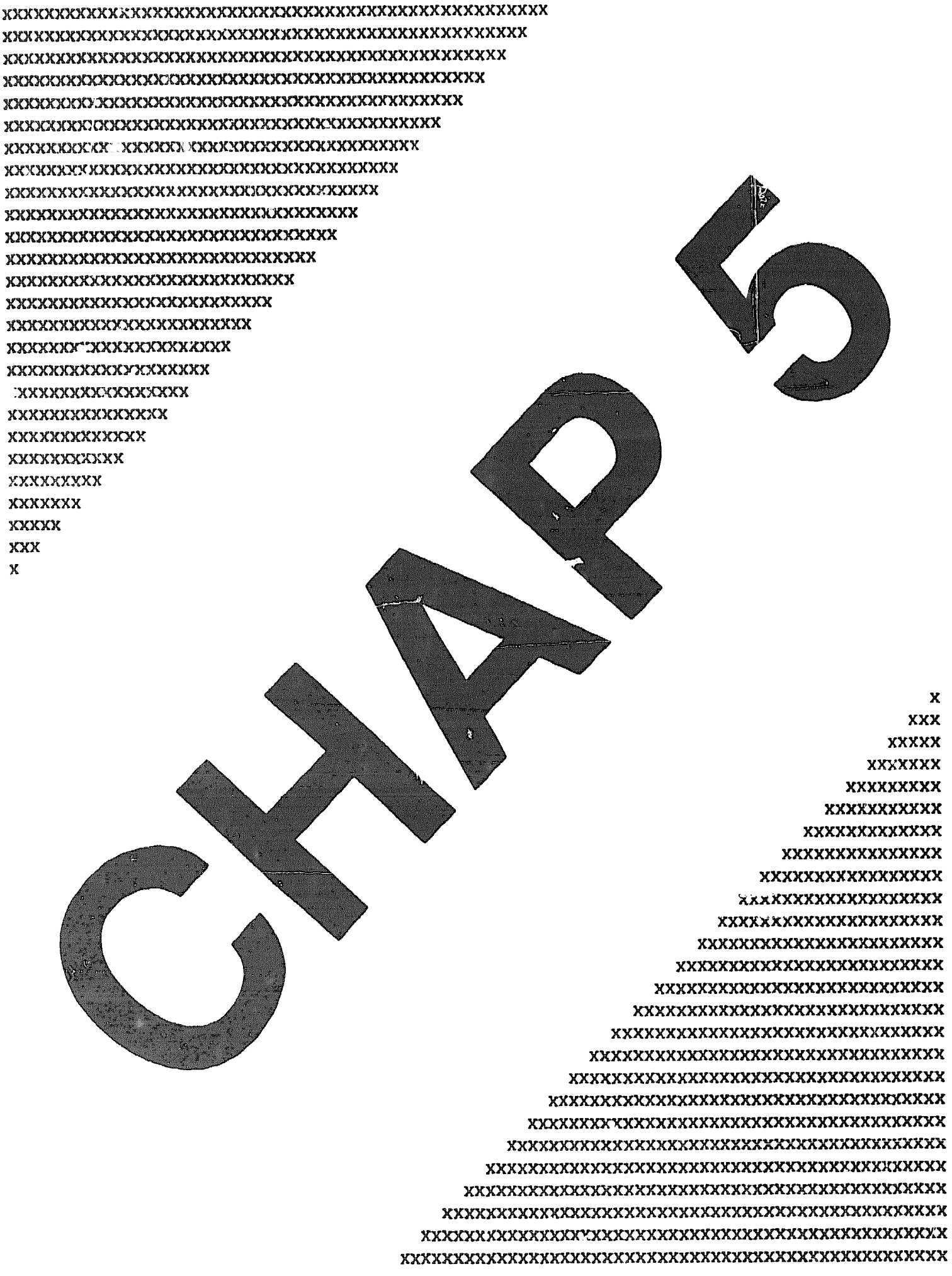
4.4.2.4 Breakpoint Message — When the program reaches a breakpoint, ODT restarts and displays a message, such as:

BK*nr*	-Where *n* is the breakpoint and *r* is the breakpoint address

Example:

BK 1 0010F0	-Displays breakpoint message
*	-Returns the ODT prompt

In the example, the program halts after reaching breakpoint 1 at location 10F0. ODT issues the asterisk prompt and waits for a command.


CHAP 5

5
Functional/Logic Description

5.1 General

This chapter is an in-depth functional description of the DECrouter 200 logic. Descriptions are written at the block diagram level and should be used with the maintenance print set, Document P/N MP01827-01.

The following is an outline of the DECrouter 200 subsystems. The router as a system is illustrated in Figure 5-1.

- Microcomputer Logic:
  - MC68000 Central Processing Unit (CPU)
  - Data/address bus
  - Power-up reset logic
  - System clock logic
  - Bus arbitrator logic
  - Interrupt control logic
  - Counters and timers
- Router Memory:
  - Address decode logic
  - 512 KB of DRAM
  - 32 KB of ROM
  - 2 KB of EEPROM

- Terminal Interface:
  - Four 2681 Dual Asynchronous Receiver/Transmitters (DUARTs)
  - Eight RS-232-C/CCITT V.24 compatible driver/receivers or one DEC423 port
- Ethernet Interface (chip set):
  - Local Area Network Controller for Ethernet (LANCE)
  - Serial Interface Adapter (SIA)
  - ESD/EOS protection circuitry
  - Octal drivers and receivers
- Modem Control
  - DC7053 chip: four modem control interrupt lines
  - DUARTs 0-3: one modem control interrupt line

5.2 Microcomputer Logic

5.2.1 CPU and Data Address Bus

For a block diagram of the DECrouter 200 implementation of the MC68000 CPU, see the print set

NOTE

The abbreviations MC68000 and CPU are synonymous throughout this chapter

The MC68000 is a 16-bit microprocessor that contains eight 32-bit data registers, eight 32-bit address registers, one of which is the special purpose stack pointer, a 32-bit program counter, an 8-bit status register, an ALU, and conditional branching logic. In the DECrouter 200, the MC68000 performs arithmetic and logic functions and handles most data and address transfers. In the status register, the low byte is the user byte, and the high byte is the system byte.

For detailed information and complete instruction sets on the CPU, see the *Motorola Microprocessors Data Manual*.

5.2.2 Device Addresses

Following are the hexadecimal addresses derived from the 20-bit address bus for the DECrouter 200 hardware subsystems.

- RAM Address 00000 - 07FFFF
- EEPROM Address 80001 - 080FFF
- DUART0 Address 100001 - 10001F
- DUART1 Address 100021 - 10003F
- DUART2 Address 100041 - 10005F

- **DUART3 Address** 100061 - 10007F
- **LANCE Address** 100080 (Data Register)
  100082 (Address Register)
- **MODEM Address** 1000A1 - 1000AB
- **CONFG REG Address** 1000C1
- **EPROM Address** 180000 - 187FFF
- **ADRPROM Address** 1C0001 - 1C003F

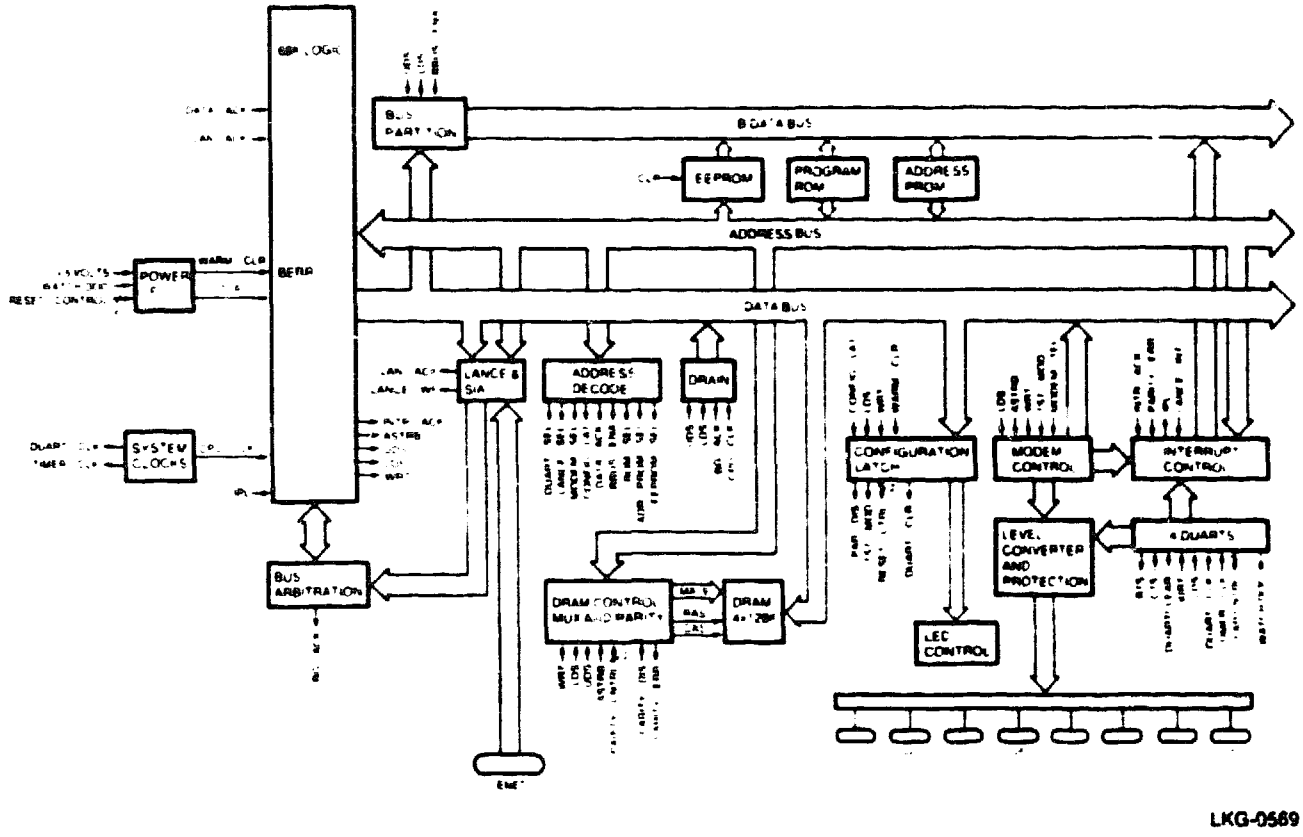


Figure 5–1: DECrouter 200 Logic

5.2.3 CPU and Data/Address Bus Signal Description

The router uses two data buses, one buffered, and one nonbuffered. The following devices are accessed through the buffered data bus:

- EEPROM
- EPROM
- ADRPROM
- DUARTs 0 through 3

The following devices are accessed through the nonbuffered data bus:

- 68000 CPU
- LANCE
- DRAM
- Configuration Latch
- Gate Array

Following are the signal names and the CPU and data/address signals in the DECrouter 200:

- *Data/Address Bus Signals*
  - **Address Bus (BUS__A01__H through BUS__A20__H)** — BUS<A20:A01> H — The router uses 20 of the 23 three-state address lines to provide direct addressing for up to two megabytes of data. These lines are drive-only for the CPU.

    The router does not use BUS__A00__H, but the CPU and the LANCE both use internal address bit <A00> to generate BUS UDSL or BUS LDS L on a byte transfer. Both signals are generated on a word transfer. The CPU uses the complement of <A00> for byte transfers. <A00> on a 1 asserts BUS LDS L; <A00> on a 0 asserts BUS UDS L.
  - **Data Bus Nonbuffered (BUS__D00__H through BUS__D15__H)** — This 16-bit, bidirectional three-state bus is the main data path for all transfers with memory and device address space.
  - **Data Bus Buffered (BBUS__D00__H through BBUS__D15__H)** — This 16-bit, bidirectional, three-state bus is used for slower devices.
- *Asynchronous Bus Control Signals*
  - **Address Strobe (ASTRB__L)** — This signal is asserted by the CPU to initiate a register transfer with an external device, such as the LANCE or a DUART. This signal also indicates that the CPU has a valid address asserted on the address bus.
  - **Read/Write (WRT__L)** — This is a wire-OR signal. WRT__L is asserted by the CPU or the LANCE to define a bus transfer.

    Negated (High) = Read
    Asserted (Low) = Write

- **Upper/Lower (Byte) Data Strobe (UDS_L/LDS_L)** — The CPU uses either or both of these signals with WRT_L to control bus transfers. Table 5–1 shows how the valid data bytes for a read or a write transfer are indicated.

Table 5–1: Data Strobe Control of Data Bus

UDS_L	LDS_L	WRT_L	Valid Data BUS_D <15:08>	Asserted on BSU_D <07:00>
0	0	-	No	No
0	1	0	No	Yes
1	0	0	Yes	No
1	1	0	Yes	Yes
1	1	1	Yes	Yes

(0 = Negated High, 1 = Asserted Low)

NOTE

As a bus slave, the LANCE ignores UDS_L and LDS_L and accepts word transfers. Wire-OR signals are driven by the LANCE and the CPU at different times and are buffered at the CPU as:

Input	Buffered Output
AS_L	BASTRB_L (LANCE not used)
WRT_L	BWRT_H,_L
UDS_L	BUDS_H,_L
LDS_L	BLDS_H,_L

- **Data Transfer Acknowledge (DTACK L)** — Asserted by the selected device in response to the ASTRB_L from the CPU. DTACK L is asserted by the following signals:

  DATA_ACK_L from the PAL — This signal is asserted during all CPU transfers with devices other than the LANCE.

  LAN_ACK_L from the LANCE — This signal is asserted during CPU transfers with LANCE registers.

- *Interrupt Control Signals*

  - **Interrupt Priority Level (IPL <2:0> L)** — This signal is asserted by the interrupt priority logic for the highest asserted interrupt request. Any asserted code, other than zero, generates a program interrupt request in the CPU.

    The CPU contains a 3-bit interrupt mask in the system byte that indicates the current processor IPL and holds an interrupt, pending whenever its priority level is higher than the requesting device. The CPU issues the interrupt when its current IPL is lower than the requesting device. The only exception is the IPL7, which allows an unmasked interrupt for the program RAM parity error.

    During the interrupt cycle, the CPU asserts the IPL of the device being serviced on address bus lines, BUS_A01–A03. The CPU asserts all other address bus lines high.

- *Bus Arbitration Signals*
  - **Bus Request (BR_L)** — This signal, asserted for the LANCE by the bus arbitrator, gains access to the data/address bus and performs a DMA transfer with program RAM.
  - **Bus Grant (BG_L)** — This CPU response to a bus request indicates that the CPU releases control of the bus at the end of the current bus cycle.
  - **Bus Grant Acknowledge (BGACK_L)** — This signal, asserted for the LANCE by the bus arbitrator, is in response to a bus grant. The LANCE assumes control of the data/address bus when the arbitrator asserts BGACK_L and LAN_HLD_ACK_L.
- *Processor Control Signals*
  - **Halt (HALT L)** — The CPU asserts this bidirectional line to external devices. When this line is driven by an external device, it causes the processor to stop at the completion of the current bus cycle.

    In the DECrouter 200, the halt and reset lines are asserted together as a system reset.
  - **Clear (CLR L)** — The CPU asserts this bidirectional line to initialize external devices with a RESET instruction. The CPU initializes when this line asserts externally.

    CLR_L is asserted on power up for a minimum of 100 milliseconds. When this signal negates, the CPU starts program execution in memory location 0, and the POWER_LED_L signal is asserted. This causes D1, on the outside of the router, to light.
  - **Function Code (FC<2:0>H)** — This signal indicates to external devices the type of reference being executed by the CPU.

    The CPU has two levels of access privileges: supervisor level and user level. Only level 7 is used by the DECrouter 200. The reference types are defined as follows:

FC<2:0> Value	Reference Type
0	-
1	User data
2	User program
3	-
4	-
5	Supervisor data
6	Supervisor program
7	Interrupt Acknowledge (INTR_ACK_L)

  - **CPU Clock (CPU_CLK_H, 10 MHz)** — The TTL-compatible input is buffered internally to create the CPU clocks.

5.2.4 Power-Up Sequencer Logic

At power up, the hardware reset signal is held high for 100 milliseconds minimum, resetting the 68000, the LANCE, the DUARTs, and other logic. Table 5–2 gives the reset signals and their functions.

Table 5–2: Reset Signals and Functions

Signal	Function
EEPROM_CLR_L	Prevents false writes to EEPROM at power up and power down.
CLR_L	Resets the module and the 68000.
WARM_CLR_L	Resets the module except the 68000 CPU, which is put into the Bus Error Exception Processing Mode. In this mode, errors are reported instead of being cleared.
POWER_FAIL_L	Reserved for use during manufacturing tests to generate power-off/power-on cycle resets.
WATCH_DOG_L	The Watchdog Timer from DUART3 generates a power-up reinitialization if not periodically cleared by the program.

5.2.5 Reset Circuitry

The reset circuitry monitors the Vcc voltage level, sensing when the module is powered up or powered down. During power changes, the circuitry generates a clear signal to the EEPROM for protection from data corruption. An RC network provides a delay to reset the CPU and the LANCE.

Hysteresis thresholds on the reset circuitry determine when a signal is asserted: a clear signal is asserted when power is applied to the module. The Vcc level then raises to 4.6 volts, and a 100-millisecond RC timeout begins. The RC timer then issues a timeout, and the clear signal is deasserted. The clear signal is then reasserted when power is removed, and the Vcc level drops below 4.5 volts.

5.2.6 Warm Reset

In normal operation a warmstart circuit allows the module to recover from unexpected processor halts and from a misaligned program counter. During warmstart reset, the memory contents are up-line loaded to a host computer, and examined for potential problems. After the up-line load, the unit starts initialization to recover from errors.

5.2.7 Reset Testing

The analog circuitry in the reset device is not tested by Self-test: the digital circuitry is tested. The digital circuitry includes a Watchdog Timer in DUART2, a flip-flop, a one-shot, and control bits from the Configuration Register. These devices generate the WARM_CLR_L signal that provides a warm reset to the module.

Self-test checks this logic by triggering the Watchdog Timer. The timer output signals the flip-flop to latch. The flip-flop output signals the one-shot to generate a 1.3 microsecond reset (WARM_CLR_L). The reset clears the circuitry except the MC68000. The WARM_CLR_L signal interrupts the CPU at the BERR_L input, causing an interrupt service routine. Self-test checks that the circuit is active, by verifying interrupt service routine execution and that the flip-flop is in its latched state.

Self-test can read the state of the flip-flop by testing the WARM_START_L signal on DUART0. Self-test can also clear the flip-flop by toggling the WARM_START_CLR_H signal on the configuration register. The RESET_DIS_H signal on the configuration register can block or can disable a reset signal while allowing testing of the flip-flop.

5.2.6 System Clock Logic

The router implements five clocks:

1. TCLK — a 10–MHz clock from the SIA
2. RCLK — a 10–MHz clock from the SIA
3. A 3.6864–MHz baud rate for the DUARTs
4. A 1.84–MHz clock to generate the DRAM refresh timer and general purpose timer
5. A refresh timer to generate the watchdog timer and the LED timer

TCLK generates two 10-MHz clocks in opposite phase of each other: CPU_CLK_H and CPU_CLK_L. Both clocks are used for the MC68000, the DRAIN, the BUS arbitration, and the address decoder circuits.

The DUART oscillator operates at a base frequency of 29.4912 MHz. A frequency divider produces the 3.6864–MHz and the 1.84-MHz frequencies.

Table 5–3 gives the functions of the three system clocks.

Table 5–3: System Clock Functions

Signal Name	Frequency	Function
CPU_CLK_H	10 MHz	Provides the time base for the CPU, the PAL, the bus arbitrator, and drain circuits
CPU_CLK_L	10 MHz	
DUART_CLK_H	3.68 MHz	Provides the transmit and receive time base for the DUARTs
TIMER_CLK_H	1.84 MHz	Provides the counter/timer clock for DUARTs 0 and 1

5.2.9 Bus Arbitration Logic

The bus arbitration logic provides communication between the LANCE and the MC68000 for bus master. A DMA transfer is initiated as follows:

- The LANCE deasserts a HOLD (BLAN__HLD__L) signal.
- The bus arbitration asserts a bus request (BR__L) signal.
- The MC68000 deasserts a bus grant (BG__L) and finishes the bus cycle.
- The MC68000 asserts an address strobe (ASTRB__L).
- The bus arbitration asserts a bus grant acknowledge (BGACK__L) to the MC68000 and a hold acknowledge to the LANCE.
- The BGACK signals the 68000 that the bus is in use, and LAN__HLD__ACK__L signals the LANCE to start a DMA.

5.2.10 Interrupt Control

The MC68000 has seven hardware interrupt priority levels (IPLs), which are expandable by external logic. Level 7 has the highest priority and level 1 the lowest. When an interrupt is not being processed, the CPU executes instructions at level 0. Priority 1 has 8 transmit levels. Each of the eight serial signals has a unique interrupt vector that is generated on a character-for-character basis. Interrupts are asserted pending the following conditions:

- The DUARTs transmitter is enabled.
- The OPCRs in the DUARTs are configured for interrupts.
- There are no characters in the transmit hold registers.

At IPL 2 there is a general timer interrupt. The interrupt is asserted pending the following conditions:

- The correct timer value is in the counter register (CTUR, CTRL) in DUART1.
- The counter timer source in the ACR is set for TIMER EXTERNAL (IP2).
- The output port (OPCR) bit 3 is set as a timer output.
- The timer is set.
- The timer expires.

An interrupt terminates when the timer restarts by reading address 10003D.

IPL 3 is the LANCE interrupt, which is asserted pending the following conditions:

- The LANCE has completed the initialization block, the transmit descriptor rings, or the receive descriptor rings.
- The INEA bit in CSR0 is set.
- One of the bits in CSR0 is set: BABL, MISS, MERR, RINT, TINT, or IDON.

The interrupt terminates when a 1 is written to the corresponding error bit in CSR0.

Priority 6 contains a refresh timer interrupt that asserts when the following parameters are met:

- The appropriate timer value is in the Counter Register CTUR/CTRL, in DUART0.
- The counter timer source in the ACR is configured for TIMER EXTERNAL — IP2.
- Output port bit 3 in the OPCR is designated as the timer output.
- The timer is active.
- The timer expires.

IPL 6 asserts every 3.8 milliseconds to refresh the Dynamic RAM. This interrupt terminates by reading address 10001D, which restarts the counter timer.

NOTE

In order to execute from DRAM: the counter timer must be active, and the interrupt address vectors and interrupt service routine must be set up.

IPL 7 holds the parity error interrupt. This interrupt asserts during a read cycle by either the CPU or by the LANCE when the parity checker finds a data error from Program RAM or from the parity memory. When a parity error generates during a LANCE memory cycle, the LANCE resets and the CPU acquires the data bus. The CPU then processes the level 7 exception. The interrupt clears when the CPU executes the interrupt acknowledge cycle for the exception.

5.2.11 Interrupt Vector Addresses

The vector addresses for the DUART transmit signals, priority 1, follow. All vectors are in hexadecimal.

TXINT0 – Vector 120

TXINT1 – Vector 124

TXINT2 – Vector 128

TXINT3 – Vector 12C

TXINT4 – Vector 130

TXINT5 – Vector 134

TXINT6 – Vector 138

TXINT7 – Vector 13C

Other vector addresses are for:

- The general use timer, priority 2,

  GENERAL_TIM – Vector 15C

- The LANCE vector, priority 3,

  LANCE_INT – Vector 17C

- The modem-control interrupts, priority 4,

  DSR_PROC_INTR – Vector 180

  CD_PROC_INTR – Vector 184

  RI_PROC_INTR – Vector 188

  CTS_PROC_INTR – Vector 18C

  SMI_PROC_INTR – Vector 190

- The DUART receive vectors, priority 5,

  RXINT0 – Vector 1A0

  RXINT1 – Vector 1A4

  RXINT2 – Vector 1A8

  RXINT3 – Vector 1AC

  RXINT4 – Vector 1B0

  RXINT5 – Vector 1B4

  RXINT6 – Vector 1B8

  RXINT7 – Vector 1BC

- The refresh timer, priority 6,

  REFRESH_TIM – Vector 1DC

- The vector for the parity error, priority 7,

  PARITY_ERR – Vector – 1FC

- The warmstart, bus error,

  WARM_ERR – Vector 8 Internal vector

5.3 Modem Control and Interrupts

Interrupt priority levels 4 and 5 are for modem control. There are eight modem interrupt signals, one for each modem. The eight interrupt signals account for 40 interrupts from five sources. Modem interrupts are only needed for input signals. Four of the interrupts are generated from a DC7053 gate array. The fifth interrupt (CTS) is generated from each of the four DUARTs. Figure 5–2 is a block diagram of the DC7053.

The modem logic latches input signals from a change detect circuit, and detects each transition of the signal. A change in state causes an interrupt. Transmitted modem control signals, except TXD, are not interrupt driven. Interrupt priority signals IP6 and IP7 are software driven.

5.3.1 Gate Array (DC7053)

The DC7053 is an 84-pin CMOS gate array. The device contains six registers: four input registers for read access and two output registers for write access. All six registers are accessed through the data bus. The input registers are dual-rank synchronizers. The clock for the synchronizers is strobed from the MC68000 every 400 nanoseconds. Each synchronizer uses a unique, modified version of address strobe. Each register generates a unique version of address strobe and a unique interrupt.

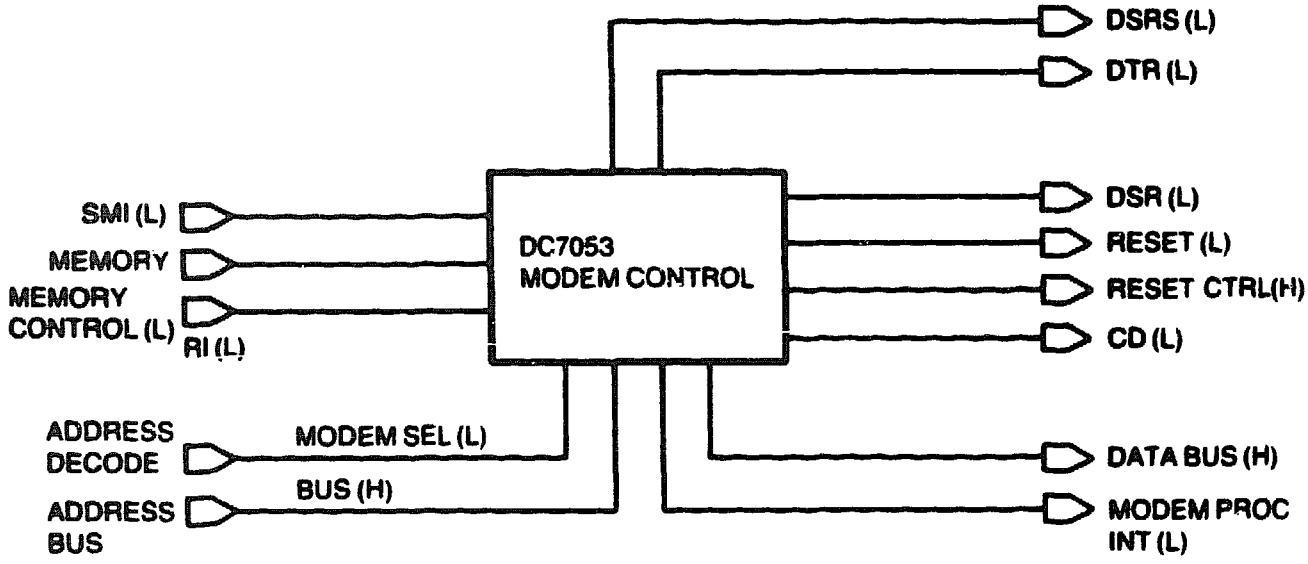


Figure 5–2: DC7053 Block Diagram

5.3.2 Register Format for Modem Control Signals

Table 5–4 shows the register formats for each of the four received modem-control signals. Each receive register produces an interrupt.

Table 5–4: Input Modem Register Formats

	BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
Data Set Ready	DSR0	DSR1	DSR2	DSR3	DSR4	DSR5	DSR6	DSR7
Carrier Detect	CD0	CD1	CD2	CD3	CD4	CD5	CD6	CD7
Ring Indicator	RI0	RI1	RI2	RI3	RI4	RI5	RI6	RI7
Speed Mode Indicator	SMI0	SMI1	SMI2	SMI3	SMI4	SMI5	SMI6	SMI7

Table 5–5 shows the register formats for the output modem-control signals.

Table 5–5: Output Modem Register Formats

	BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7
Data Terminal Ready	DTR0	DTR1	DTR2	DTR3	DTR4	DTR5	DTR6	DTR7
Data Signal Rate Select	DSRS0	DSRS1	DSRS2	DSRS3	DSRS4	DSRS5	DSRS6	DSRS7

5.3.3 Modem Connection

Figure 5–3 shows the hardware configuration for the connection of data terminal equipment (DTE) to data communications equipment (DCE) that is standard for modem use. The signals in Figure 5–3 function in the DECrouter 200 as listed in Table 5–6.

Table 5–6: DTE-to-DCE Connection Signals

Signal Name	DECrouter 200 Function
TXD/RXD	Indicates data exchange
RTS/CTS	Indicates continuous activity/programmable
DSR/DTR	Indicates a device is on line
CD	Indicates on line activity
RI	Indicates incoming call/automatic answer
LL	Establishes a local loop test with the DCE
RL	Establishes a remote loop test with the DCE
DSR/SMI	Indicates switch between high-speed and low-speed for devices using specified option

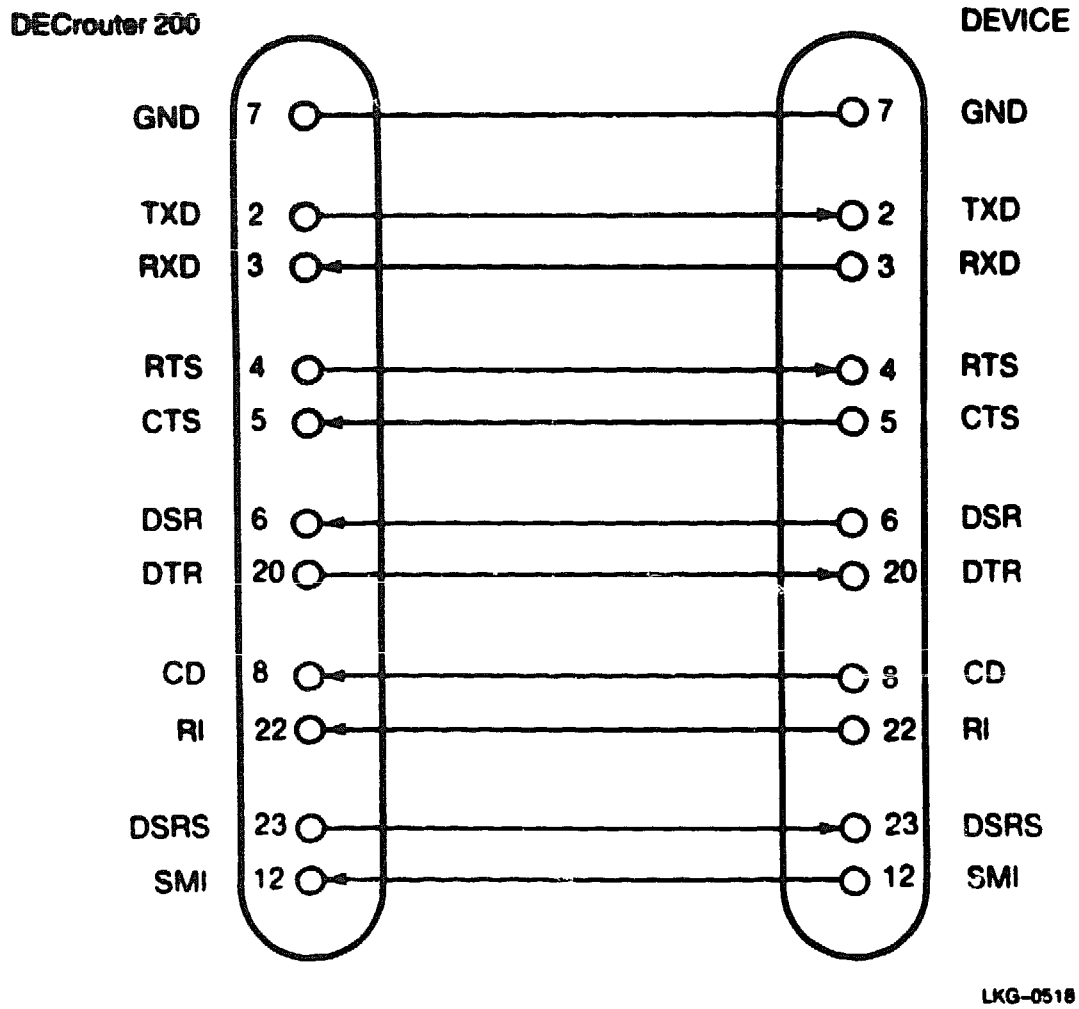


Figure 5–3: DTE-to-DCE Hardware Configuration

5.3.4 Null Modem Connection

Figure 5–4 shows the hardware configuration for a DTE-to-DTE connection to use the DECrouter 200 as a virtual DCE. Table 5–7 shows the signals in a DTE-to-DTE connection.

Table 5–7: DTE-to-DTE Connection Signals

Signal Name	DECrouter 200 Function
TXD/RXD	Indicates data exchange
RTS/CTS	Indicates flow control
RTS	Is controlled functionally as CTS in software
CTS	Is controlled functionally as RTS in software
DSR/DTR	Indicates flow control or on-line device
DSR	Is controlled functionally as DTR in software
DTR	Is controlled functionally as DSR in software
CD	Is ignored
RI	Is ignored
LL	Is controlled functionally as RI, establishing calls as a DCE
RL	Is controlled functionally as CD, indicating an active network
DSRS/SMI	Indicates a switch between high-speed and low-speed for devices using specified option

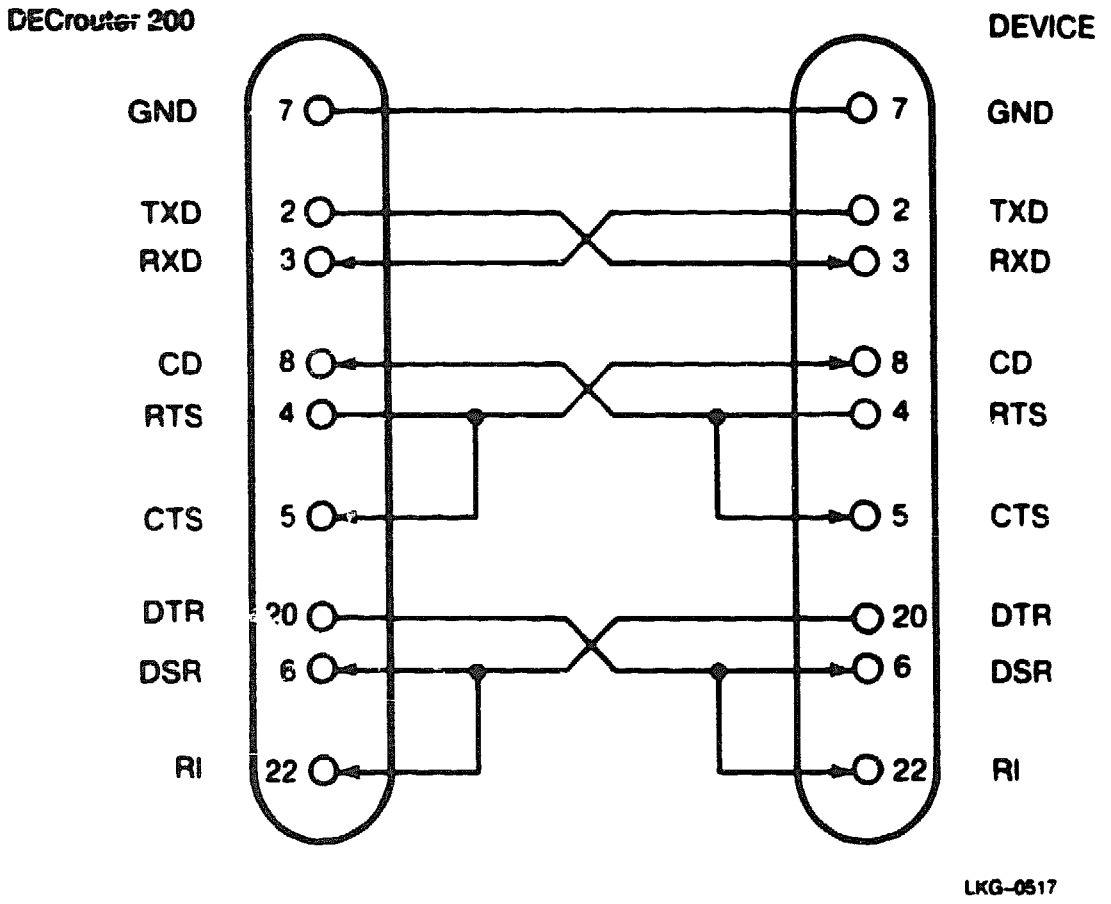


Figure 5–4: Null Modem Hardware Configuration

5.4 Router Memory Subsystem

The addressing logic in the DECrouter 200 allows the CPU, under program control, to access the following:

- The DUART and the LANCE internal registers
- 512 KB of dynamic program RAM
- 32 KB of PROM
- 2 KB of EEPROM
- 32 bytes of physical address PROM (PA PROM)

The addressing logic also allows the LANCE to perform direct memory access (DMA) transfers with RAM. Figure 5–5 illustrates memory allocation in the DECrouter 200.

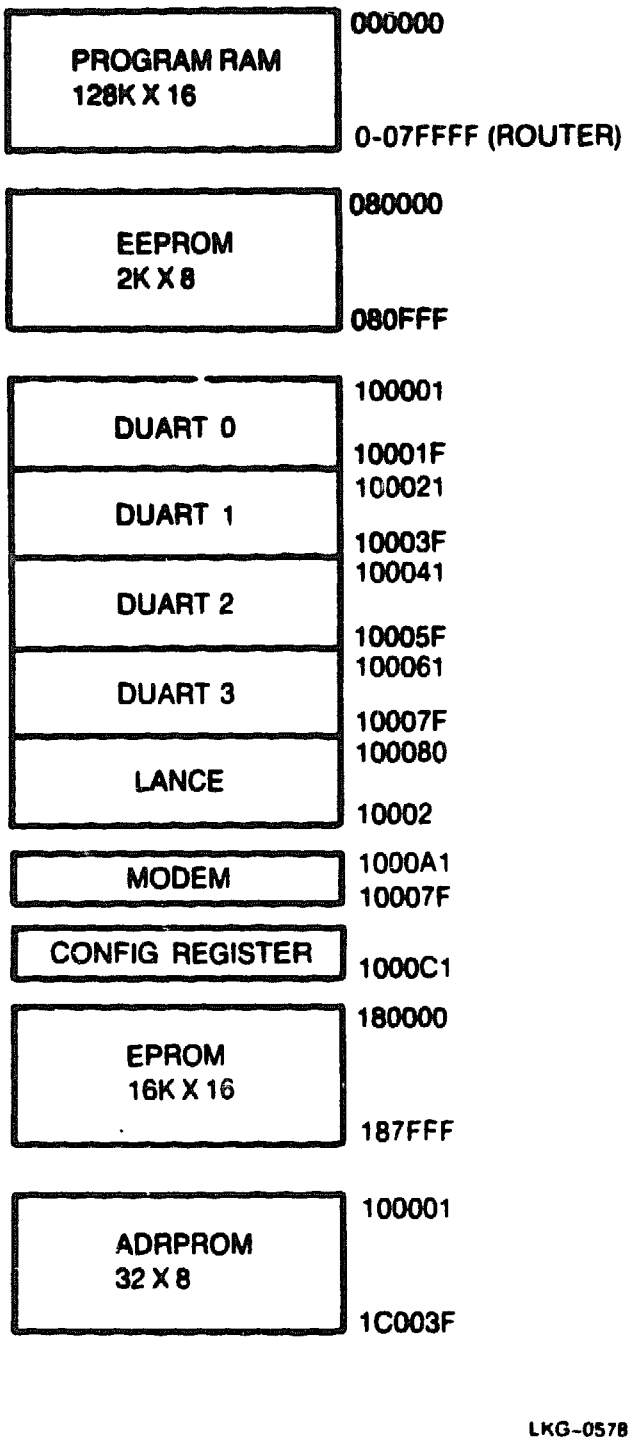


Figure 5–5: DECrouter 200 Address Space Allocation

5.4.1 Address Selection

Asynchronous device address space is selected by the program address bit, BUS <A20:A00>. Figure 5–6 shows the CPU selection of the DECrouter 200 address space.

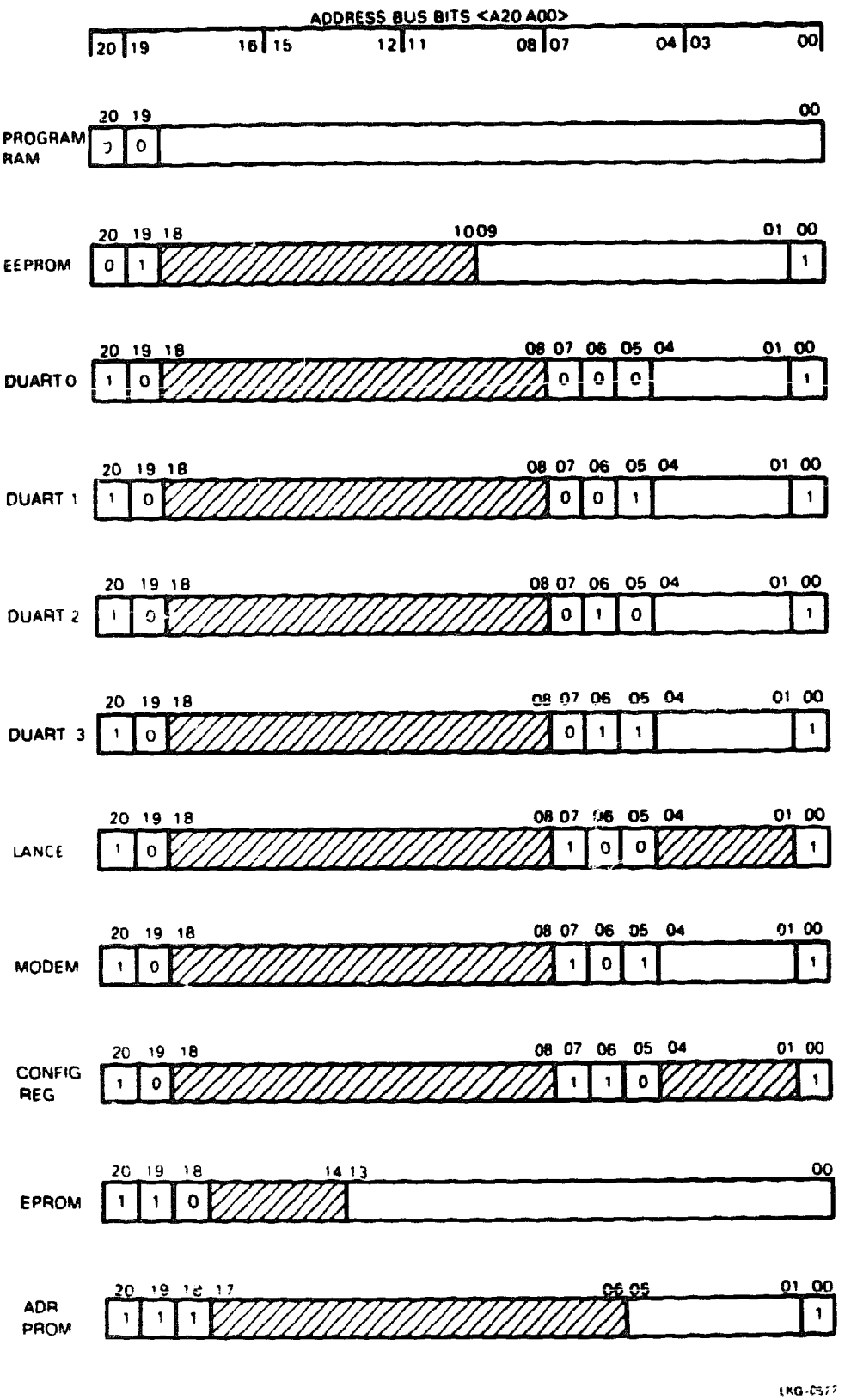


Figure 5–6: CPU Selection of Router Address Space

The address decoder contains a fusable-link array device called Programmable Array Logic (PAL), programmed as shown in Table 5–8.

Table 5–8: Programmable Array Logic Functions

Signal	Device Selected
ROM_SEL_L	Program ROM
EEROM_SEL_L	EEPROM
ADR_PROM_SEL_L	Ethernet Address
DUART_SEL_L	DUART Register
LANCE_SEL_L	LANCE Slave
MODEM_SEL_L	DC7053 Gate Array
CONFIG_LAT_L	Configuration Register
DATA_ACK_L	Acknowledge Memory Cycle
PAL_STATE_2_L	Enable Writes to EEPROM
BBUS_ENA_L	Enable Bus Transceivers

NOTE

The Enable Bus Transceivers signal is for access of data over the buffered data bus. RAM addresses are not selected with the PAL. The RAM logic has an internal address decoder.

The CPU uses the complement of address bit <A00> on a byte transfer.

Also:

<A00> on 0 = Selects data bits BUS DAL<15:08>

<A00> on 1 = Selects data bits BUS DAL<07:00>

As bus slave, the LANCE is accessible only on word boundaries. As bus master, LANCE uses the standard Digital convention in selecting the upper or lower byte, which is:

<A00> on 0 = Selects data bits BUS DAL<07:00>

<A00> on 1 = Selects data bits BUS DAL<15:08>

Only program ROM and program RAM locations are selected on the upper or lower byte. All other devices are accessible only on the lower byte.

5.4.2 Power-Up Addressing

When the signal clear (CLR_L) is negated following the power-up sequence, the CPU fetches its program counter (PC) and stack pointer (SP) from memory location 0.

Location 0 is usually defined in program RAM address space. But the PAL logically swaps the program ROM and program RAM address space, selecting program ROM under the following power-up conditions:

1. Output port 2 of all four DUARTs asserts the complement of output-port register bit OPR<2> as 0, active low.
2. DRAM_ENA_L, OP2 from the fourth DUART, is asserted high to the PAL.
3. Location 4 of Program ROM points to the Self-test power-up routine. RAM_ENA_L is then negated or asserted under program control.

Self-test starts program execution in ROM. Self-test enables RAM, and starts executing tests in sequence. Some test routines are copied to, and operate in RAM. Self-test then jumps to the initialize program, which transmits a request for a down-line load of router operating software from a load host volunteer. Chapter 3 provides a detailed explanation of down-line loading.

5.4.3 DUART and LANCE Register Addressing

The CPU selects the DUART and the LANCE internal registers by asserting address bus bits BUS <A20:A01>, as follows:

1. With BUS <A20:A19> H equal to 10, the PAL asserts DUART/LANCE L, enabling the address decoder.
2. BUS <A07:A05> to the address decoder selects a DUART or the LANCE by asserting one of the signals PJC ENB <3:0> L or PJC ENB LANCE L.
3. DUART register selects BUS <A04:A01> for a read or a write transfer.
4. The LANCE internal latching Register Address Port (RAP) selects one of four Control and Status Registers (CSRs).

5.4.4 Program Random Access Memory

The router uses up to 512 KB of dynamic RAM as the operating main memory from where the router operating software is executed. Program RAM uses a multiplexed 9-input row and column-addressing method selected and clocked by a delay line as shown in the print set.

5.4.5 CPU-Initiated Transfer

The CPU starts a transfer with program RAM as follows:

1. The CPU asserts an address on BUS<A20:A21> and asserts buffered address strobe (BASTRB_L).
2. With BUS<A20:A19> equal to 00, combinational logic circuitry asserts RAM_ENA_L, enabling the signal input to the delay line.

5.4.6 LANCE-Initiated Transfer

The LANCE starts a transfer with program RAM as follows:

1. When the LANCE gains access to the bus, the arbitrator asserts LAN_HLD_ACK_L, enabling the LANCE input to the delay line.

2. The LANCE asserts an address on address BUS<A16:A01> and asserts address latch enable (ALE_H, ASTRB_L and ADR_ENA_H), which opens the address latch for BUS<A15:A01>.

5.4.7 Data Transfer Cycle

The data cycle of a DMA transfer initiated by the CPU or the LANCE follows this sequence:

1. A buffered write (BWRT_L) defines the transfer as a read or a write.
2. When the CPU asserts buffered address strobe (BASTRB_L), or the LANCE deasserts ALE_H, the high-to-low transition to the delay line generates row address strobe (HBT_RAS_L and LBT_RAS_L).
3. On a low-to-high transition on the delay line input, the row and column addresses are loaded to program RAM, selecting the transfer address as follows:
   - The address value on BUS<A08:A01> is asserted on MA1 to MA9 from the address multiplexer and is clocked to the program RAM row address register by HBT_RAS_L and LBT_RAS_L.
   - When RAM acknowledge (RAM_ACK_H) is asserted by the delay line, the address value on BUS<A16:A09> is asserted on MA1 to MA9 from the address multiplexer.

5.4.8 Dynamic RAM

There are four 128-KB banks in DRAM. Any bank can be depopulated to decrease its memory size. There are also two 256 K X 1 memory chips, used for parity, that cannot be depopulated.

Dynamic RAM requires refresh, and causes the router software to interrupt every 3.8 milliseconds. During the interrupt, the CPU reads 512 consecutive bytes of DRAM, updating all the memory cells in the four banks of RAM and the parity chips.

Dynamic RAM has two parity bits for each word. The low-byte (D0 to D7), has even parity, and the high-byte (D8 to D15) has odd parity. Write cycles generate and store parity, while read cycles interrupt the CPU when errors are read.

NOTE

- The router software initializes all DRAM on power up because reading unused memory locations may indicate parity errors.
- The write cycle generates the parity, and the read cycle tests parity that is written by the write cycle.

Parity error interrupts occur on IPL 7, are always fatal, and are not masked by the software. The software can prevent interrupts by completely disabling parity. The software can also force parity errors by changing bit states on the DUART output ports, which enables testing of the parity logic.

When the CPU or the LANCE asserts BUS UDS L and/or BUS LDS L, then MA1 to MA9 are clocked to the RAM column address register by either or both of these signals:

- Upperbyte column address strobe (HBT_CAS_L)

- Lowerbyte column address strobe (LBT__CAS__L)

On a read, the program RAM data lines are enabled as data outputs by HBT__CAS__L and/or LBT__CAS__L.

5.4.9 Erasable Program ROM (EPROM)

The router uses 32 KB of EPROM for storing the loader for the Load/Dump code. The EPROM is also used for Self-test, for diagnostics, and for the Online Debugging Tool (ODT).

5.4.10 Physical Address Programmable ROM (PA PROM)

Every router has a unique physical Ethernet address in 32 bytes of PA PROM.

The address PROM locations are selected as follows:

1. With BUS<A20:A19> equal to 11 and BUS<A18> equal to 1, the PAL circuit asserts ADR__PROM__SEL__L and selects the physical address PROM.
2. BUS<A05:A01> then selects one of 32 byte locations in the low-byte position. Bus selection is enabled by BLDS__L.

5.4.11 Electrically Erasable Programmable ROM (EEPROM)

The router has 2 KB of nonvolatile EEPROM in which to store the terminal default, the modem-control, and the system default parameters for each communication line. These parameters can be changed by a user and stored either permanently in EEPROM or temporarily in DRAM. During power up and power down, the reset circuit protects the EEPROM from data corruption by asserting the CTRL signal, which holds the EEPROM Output Enable input low.

5.4.12 Write Inhibit

The EEPROM circuit has active low inputs and output signals are low. The write protect bit should be read as 1 to verify that the EEPROM writes are disabled. This bit can be read from input port bit 5 (IP5) of DUART0.

The EEPROM internal write process takes up to 10 milliseconds. The address, the data, and the control signals are internally latched by the EEPROM, and the processor write cycle takes 600 ns. This allows the CPU to execute other programs during the internal write process. During a write cycle, read cycles yield random data. The read/write address range for EEPROM is 080001 to 080FFF.

The EEPROM locations are selected as follows:

1. With BUS<A20:A19> equal to 01, the PAL asserts EEPROM__SEL__L, which selects EEPROM.
2. BUS<A11:A01> selects one of 2,000 byte locations in the lower byte position. Selection of a byte location is enabled by BLDS__L.

5.4.13 Asynchronous Interface

The four DUART chips used for external interface are operated under program control through interrupts. The CPU selects the DUART internal registers on address bus bits BUS<A20:A19> and BUS<A07:A05>.

In the program, address bit <A00> must be set to select the lower byte for data transfer. BUS<A04:A01> then selects the hexadecimal address to read or to write. Table 5–9 gives the DUART registers, the hexadecimal addresses, and the read/write functions. A dash (-) means function undefined.

Table 5–9: Summary of DUART Register Functions

Bus Address	Read Function	Write Function
100001	Mode Register A	Mode Register A
100003	Status Register A	Clock Select Register A
100005	-	Command Register A
100007	RX Holding Register A	TX Holding Register A
100009	Input Port Change Register	Auxiliary Control Register
10000B	Interrupt Status Register	Interrupt Mask Register
10000D	Counter/Timer Upper Register	Counter/Timer Upper Register
10000F	Counter/Timer Lower Register	Counter/Timer Lower Register
100011	Mode Register B	Mode Register B
100013	Status Register B	Clock Select Register B
100015	-	Command Register B
100017	RX Holding Register B	TX Holding Register B
100019	-	-
10001B	Input Port State Register	Output Port Configuration Register
10001D	Start Counter/Timer Command	Set Output Port Register Bits
10001F	Stop Counter/Timer Command	Clear Output Port Register Bits

NOTE

The addresses in Table 5–9 select DUART0. To select DUART1, an offset of 20 must be added. To select DUART2, an offset of 40 must be added. To select DUART3, an offset of 60 must be added.

5.4.14 DUART Signal Descriptions

The following list describes the router implementation of the DUART operating signals.

Data/Address Bus Signals

- **Address Bus (BUS<A04:A01>H)** — The internal DUART registers are selected by the CPU with four of the three-state address bus lines. These are receive-only lines for the DUARTs that can only be bus slaves.
- **Data Bus (BUS DAL<07:00> H)** — The lower byte of the bidirectional three-state data bus is used to transfer internal register data on a CPU-initiated transfer with a DUART.

Asynchronous Bus Control Signals

- **Enable *n* (DUART*n*__SEL__L)** — Selects one of four DUARTs on a CPU-initiated transfer, where *n* is the DUART number, 0 to 3.
- **Write Enable (DUART__WRT__L)** — Enables the input of the selected register to receive write data from the data bus.
- **Read Enable (DUART__RD__L)** — Enables the three-state data bus output and asserts read data from the selected register.

Channels A and B Receive and Transmit Signals

- **Receive Data *n* (RDX*n*__L)** — Accepts serial data from the receive-level converters, where *n* is the asynchronous port number, 0 to 7.
- **Transmit Data *n* (TXD*n*__L)** — Applies serial data to the transmit-level converters, where *n* is the asynchronous port number 0 to 7.
- **Clock Input X1 (DUART__CLK__L 3.68 MHz)** — Provides the base clock (16X) from which the transmit and receive baud rates are derived for each channel.

Output Port Signals

- **Output Ports OP6/OP7 (TXINT*n*__L)** — OP6 and OP7 assert a transmit interrupt request from channels A and B (TX RDY A and TX RDY B) to the priority encoder, where *n* is the asynchronous port number, 0 to 7.
- **Output Ports OP4, OP5 (RXINT*n*__L)** — OP4 and OP5 assert a receive interrupt request from channels A and B (RX RDY A and RX RDY B) to the priority encoder, where *n* is the asynchronous port number, 0 to 7.
- **Output Port OP3** — OP3 asserts the following counter/timer signals from each DUART:
  - **DUART0** — REFRESH__TIM__L from the refresh timer interrupts the CPU to refresh the dynamic program RAM. This signal is also used to clock the watchdog timer and the LED timer.
  - **DUART1** — GENERAL__TIM__L from the counter/timer is used as the slot timer by the program.
  - **DUART2** — WATCH__DOG__L from the watchdog timer interrupts the CPU to reinitialize the router and to reload the router software if the timer is not periodically cleared by the program.
  - **DUART3** — LED__H from the counter/timer activates, by blinking, the LED indicator on nonfatal Self-test errors and leaves the LED lit for a no-error condition.

- **Output Port OP2** — OP2 asserts selection signals from each DUART:
  - **DUART0** — SET_HBT_PAR_L forces parity errors on data bits D08 to D15.
  - **DUART1** — EEPROM_WRT_H enables writing to EEPROM.
  - **DUART2** — SET_LBT_PAR_L forces parity errors on data bits D00 to D07.
  - **DUART3** — DRAM_ENA_L selects program ROM on a power up by swapping program ROM and RAM address space.
- **Output Port OP0 and OP1** — OP0 and OP1 are used for generating the RTS*n*_L signals, where *n* is the port number.

Input Port Signals

- **Input Port IP2** — IP2 in each DUART accepts the following signals:
  - **DUART0** — TIMER_CLK_H, 1.84 MHz H is the refresh timer clock.
  - **DUART1** — TIMER_CLK_H, 1.84 MHz H is the general purpose timer clock.
  - **DUART2** — REFRESH_TIM_L is the watchdog timer clock.
  - **DUART3** — REFRESH_TIM_L is the LED timer clock.

Reset Signal

- The DUARTs are reset when the configuration latch is reset and the DUART reset signal is disabled on the first write to the latch. The DUARTs remain in reset mode until the first write is done. The reset signal:
  - Clears the internal registers
  - Stops the counter/timer
  - Clears the output ports OP<7:0>
  - Sets channels A and B to inactive states
  - Sets channels A and B output to active high

Each DUART provides the following three sets of function registers:

- **Receive and Transmit Registers** — Each DUART provides two full-duplex serial asynchronous channels; channels A and B.

  The receive and transmit registers for channels A and B are described in the following section

  Each channel connects to an approved asynchronous device through transmit (TX) and receive (RX) level converters that conform to EIA RS-232-C and to CCITT V.24 specifications.
- **Input and Output Port Registers** — The input/output ports and registers are used for general functions or are configured for specific I/O functions.

- **Interrupt Control and Counter/Timer Registers** — The interrupt control and counter/timer registers are described in Section 5.2.10.

5.4.15 Receive and Transmit Registers

Figure 5–7 shows the register formats, the hexadecimal addresses, and the read/write functions for channels A and B.

The register functions are the same for both channels. The mnemonics for the bits start with the letter A or B to identify the channel.

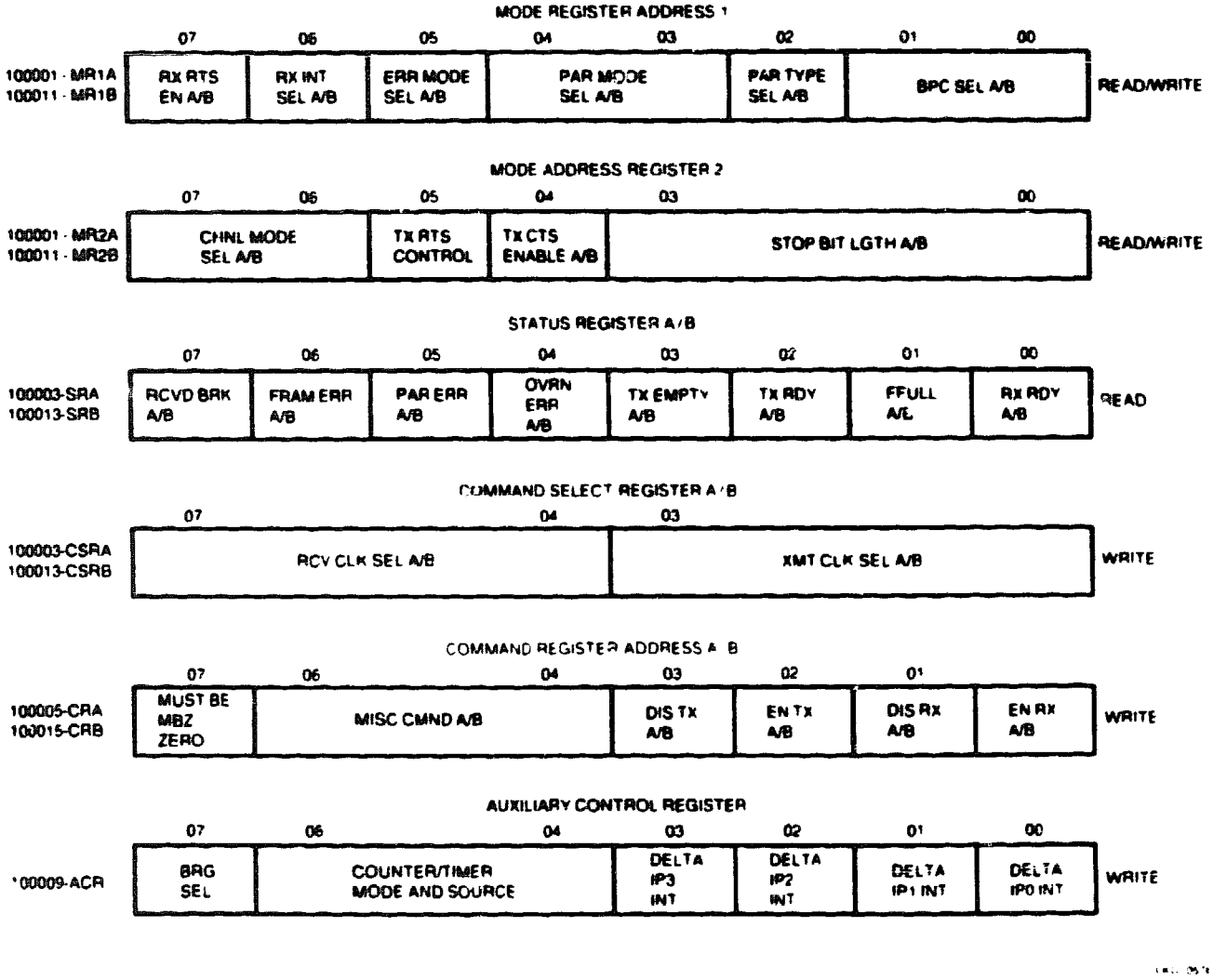


Figure 5–7: DUART Receive and Transmit Register Formats

5.5 MRnA and MRnB Mode Registers

Both channels contain two mode registers that are designated and are selected as:

Channel A — Mode Registers 1 and 2; MR1A and MR2A
Channel B — Mode Registers 1 and 2; MR1B and MR2B

Register Pointer

MR1A and MR1B are both selected by a pointer that is cleared on a power-up sequence. When MR1n is accessed for either channel, the pointer is set, selecting MR2n for all subsequent reads or writes to the channel address. The pointer remains set until the router is initialized or the Reset Pointer command is issued from the command register for that channel. MR2A is pointed to after reading or writing MR1A.

5.5.1 MR1A and MR1B Bit Assignments

Table 5–10 describes the read/write bit assignments for MR1A and MR1B, and Table 5–11 describes the bit assignments for MR2A and MR2B:

Table 5–10: MR1A and MR1B Bit Assignments

Bits	Name	Description
<07>	Receive Request to Send Enable (RX RTS EN A/B)	With bit set, output ports OP0 and OP1 assert these states: OP0 = RX RTS A OP1 = RX RTS B Each output is negated on a full FIFO buffer and is asserted when an empty FIFO location becomes available for the channel Outputs OP0 and OP1 can also be enabled for other functions
<06>	Receive Interrupt Select (RX INT SEL A/B)	Enables ISR <05> and <01> to assert either the RX RDY or the FFULL flag as follows: ISR BIT = RX INT SEL A/B Bit On a 0 / On a 1 ISR <01> = RX RDY A / FFULL A ISR <05> = RX RDY B / FFULL B
<05>	Error Mode Select (ERR MODE SEL A/B)	Selects the error mode as: 0 – Character 1 – Block
<04:03>	Parity Mode Select (PAR TYPE SEL A/B)	Selects the parity check mode as: 0 – With Parity 1 – Force Parity 2 – No Parity 3 – Multidrop
<02>	Parity Type Select (SPC SEL A/B)	Selects the parity type as: 0 – 5 bits 1 – 6 bits 2 – 7 bits 3 – 8 bits

Table 5–11: Mode Register MR2A and MR2B Bit Assignments

Bits	Name	Description
<07:06>	Channel Mode Select (HNL MODE SEL A/B)	Selects the transmit mode as follows: 0 – Normal (full duplex) 1 – Auto Echo (transmitter uses receiver clock) 2 – Local Loopback (receiver uses transmitter clock) 3 – Remote Loopback (transmitter)
<05>	Transmit Request to Send Select (TX RTS SEL A/B)	When bit set, output ports OP0 and OP1 assert the following states: OP0 = TX RTS A OP1 = TX RTS B
<04>	Transmit Clear to Send Select (TX CTS SEL A/B)	When bit is set, output ports OP0 and OP1 assert the following states: OP0 = TX CTS A OP1 = TX CTS B
<03:00>	Stop Bit Length (STOP BIT LGTH A/B)	Sets the length of the stop bit in bit-time increments. The DECrouter 200 supports the following stop bit values: 7 – 1.000 F – 2.000 The vendor specification provides a complete list of the bit time values.

5.5.2 SRA and SRB Status Registers

Table 5–12 describes the status flag and the read functions for SRA and SRB

Table 5–12: Status Register SRA and SRB Bit Assignment

Bits	Name	Description
<07>	Received Break (RCVD BRK A/B)	Indicates that an all-zero character was received without a stop bit, using only one FIFO location. This bit set also enables BRK B CHNG or the CHNG BRK A flag in ISR <06> or <02>.
<06>	Framing Error (FRAM ERR A/B)	Indicates that a stop bit was not detected when a received character was clocked to the FIFO.
<05>	Parity Error (PAR ERR A/B)	Indicates incorrect parity on a received character when the With Parity or Force Parity mode is enabled (see MR1A and MR1B <04:03>.
<04>	Overrun Error (OVRN ERR A/B)	Indicates that a character was received with a full FIFO and input character buffer, data has been lost.
<03>	Transmit Empty (TX EMPTY A/B)	Set after the last stop bit of a character is shifted from the serial transmit register when no other character is available in THRA or THRB. The bit is cleared when the holding register is loaded or when the transmitter is disabled.
<02>	Transmit Ready (TX RDY A/B)	Indicates that THRA or THRB is empty and is ready to accept another character for transmitting. The bit is set when the transmitter is enabled, and is cleared when the holding register is loaded or the transmitter is disabled.
<01>	FIFO Full (FULL A/B)	Set when a third received character is loaded to the FIFO buffer for RHRA or RHRB. The bit is cleared when a character is read, unless another bit is available in the receive shift register.
<00>	Receive Ready (RX RDY A/B)	Indicates that a received character is clocked to the FIFO for RHRA or RHRB. The bit is cleared when all characters are read.

5.5.3 CSRA and CSRB Clock Select Registers

The information in Tables 5–13 and 5–14 explains the write functions for CSRA and CSRB.

Each register selects receive and transmit baud rates for its channel from one of two sources, depending on the state of the auxiliary control register bit: ACR<07>.

Table 5–13: Clock Select Register CSRA and CSRB Bit Assignments

Bits	Name	Description
<07:04>	Receive Clock Select (RCV CLK SEL A/B)	Sets the receive clock for the channel to one of the baud rates listed below.
<03:00>	Transmit Clock Select (XMT CLK SEL A/B)	Sets the transmit clock for the channel to one of the baud rates listed below.

Two ranges of baud rates are selected for either channel by the following states of ACR<07>:

RCV/XMT CLK SEL	Set 0 (ACR<07> equals 0)	SET 1 (ACR<07> equals 1)
0	50	75
1	110	110
2	134.5	134.5
3	200	150
4	300	300
5	600	600
6	1200	1200
7	1050	1050
8	2400	2400
9	4800	4800
A	7200	7200
B	9600	9600
C	38400	19200
D	Timer	Timer
E	IP<6:3>-16X	IP<6:3>-16X
F	IP<6:3>-1X	IP<6:3>-1X

Table 5–14 shows the bit settings in clock select registers A and B.

Table 5–14: Clock Select Registers A and B

Bits	Name	Description
<07>	Must Be Zero (MBZ)	Unused and must be zero
<06:04>	MISC CMND A/B	Writing this field with a 3-bit code issues one of the following commands to the channel: 0 – No Command 1 – Reset Pointer (selects MR1A or MR1B) 2 – Reset Receiver 3 – Reset Transmitter 4 – Reset Error Status 5 – Reset Break Change Interrupt 6 – Start Break 7 – Stop Break
<03>	DIS TX A/B	Disables the channel A or B transmitter. Resets the TX EMPTY and TX RDY status bits, SRA or SRB<03:02>, see Table 5–12.
<02>	EN TX A/B	Enables the channel A or B transmitter. Sets the TX RDY status bit, SRA or SRB<02>, see Table 5–12.
<01>	DIS RX A/B	Disables the channel A or B receiver, except for the multidrop mode MR1A or MR1B <04:03>, see Table 5–10.
<00>	EN RX A/B	Enables the channel A or B receiver. The DUART special wake-up mode is not used in the DECrouter 200 configuration.

5.5.4 RHRA and RHRB Receive Holding Registers

Registers RHRA and RHRB are Receive Data Read registers.

FIFO Buffer – Both RHRA and RHRB have a three-byte FIFO that acts as a cache for up to four characters, including the serial receive input data buffer. Each FIFO location also stores the received character status: parity error, framing error, and received break.

The RX RDY A or the RX RDY B bit is set, SRA or SRB bit <00>, when one to four characters are available, and remains set until the FIFO is empty.

The FFULL A or FFULL B status bit is set, SRA or SRB bit <01>, if all three locations for the channel are full. Reading a character empties a FIFO location and negates FFULL, unless a character is waiting in the serial input buffer.

RX RDY or FFULL can be used to assert a program interrupt request. RX RDY is used in the DECrouter 200.

5.5.5 THRA and THRB Transmit Holding Registers

Registers THRA and THRB are for transmit data write functions. When a register is loaded by the CPU, the byte is parallel loaded to the Transmit Shift Register from where the byte is shifted out as serial data to a device.

5.5.6 Auxiliary Control Register ACR<07>

Table 5–15 describes the write functions for ACR bit ACR<07>, which is used with CSRA and CSRB.

Table 5–15: Auxiliary Control Register ACR<07>

Bits	Name	Description
<07>	Baud Rate Generator Select BRG SEL	Selects one of two transmit and receive baud rate ranges as described for CSRA and CSRB bit fields <07:04> and <03:00>.
<06:04>	-	Used with the counter/timer register.
03:00	-	Used with the input port change register.

5.5.7 Input and Output Port Registers

Figure 5–8 shows register settings, hexadecimal addresses, and read/write functions for the Input Port (IP) and Output Port (OP) registers.

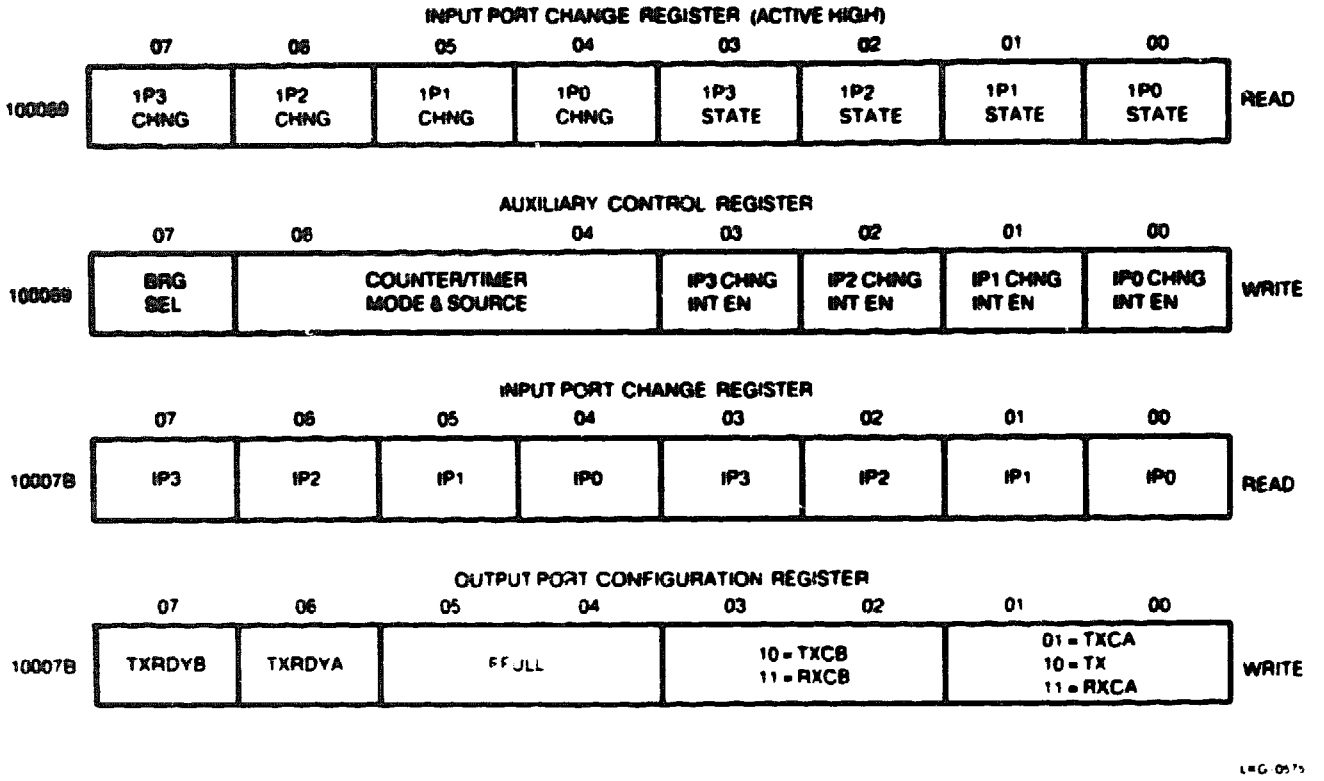


Figure 5–8: DUART Input and Output Port Register Bit Settings

5.5.8 Input Port Change Register (IPCR)

Table 5–16 describes the read functions for the IPCR.

Table 5–16: IPCR Port Change Register Bit Settings

Bits	Name	Description
<07:04>	IP<03>CHNG	A bit is set by a signal change on corresponding input port. All bits are cleared, including ISR<07> when register is read at address 4.
03:00	IP<3:0>STATE	These bits assert the states of signals on input ports IP<03>

5.5.9 Auxiliary Control Register (ACR<03:00>)

Table 5–17 describes the write functions for ACR<03:00> that are used with the IPCR.

Table 5–17: Auxiliary Control Register

Bits	Name	Description
<07>	–	Used with the clock select registers.
<06:04>	–	Used with the counter/timer register.
<03:00>	Input Port <3:0> Change Interrupt Enable (IP <3:0> CHNG INT EN)	When a bit is set, and the corresponding bit is set in IPCR <07:04>, ISR <07> is also set. The INTR output is also asserted high, and is unused in the router.

5.5.10 Input Port Status Register (IPSR)

The signal states connected to input ports IP<6:0> are read bits IPSR<06:00>; IP<7> is not used and IPSR<07> is always read as a 1.

5.5.11 Output Port Configuration Register (OPCR)

Table 5–18 describes the write functions for the OPCR, which controls the function of output ports OP<7:0>.

Table 5–18: Output Port Configuration Register (OPCR)

Bits	Name	Description
<07:04>	Output Port<7:4> Select (OP<7:4>SEL)	Control output ports OP<7:4> in one of the following two ways. Cleared: Each clear bit enables the complement of the corresponding OPR bit to output port OP<7:4>. Set: Each set bit causes the corresponding output port to assert the following signal: OP7 – Asserts channel B transmit interrupt request, the complement of TX RDY B. OP6 – Asserts channel A transmit interrupt request, the complement of TX RDY A. OP5 – Asserts channel B receive interrupt request, the complement of RX RDY B. OP4 – Asserts channel A receive interrupt request, the complement of RX RDY A.

Table 5–18 (cont.): Output Port Configuration Register (OPCR)

Bits	Name	Description
<03:02>	Output Port 2 Select (OP3 SEL)	Enables output port OP3 or OP<3:2> to assert one of the following signals:
		0 – Output ports OP3 and OP2 assert the complement of output port register bits <03:02>.
		1 – Asserts the counter/timer output at the programmed frequency in timer mode, or in overflow counter mode.
<03:02>		2 – Asserts the 1X clock output for the channel B transmitter.
		3 – Asserts the 1X clock output for the channel B receiver.
<01:00>	Output Port 2 Select (OP2 SEL)	Enables output port OP2, or OP1 and OP0, to assert the following signals:
		0 – Output ports OP<1:0> assert the complement of output port register bits OPR<01:00>.
		1 – Asserts the 16X character clock output for the channel A transmitter as follows:
		CSRA<3:0> = E, assert 16X character clock CSRA<3:0> = F, assert 1X character clock
		2 – Asserts the 1X clock output for the channel A transmitter
		3 – Asserts the 1X clock output for the channel A receiver.

5.5.12 Output Port Register (OPR)

The OPR can be used as a general purpose register or as a control register. The register is written with ones at either of two addresses that set or that clear the selected bits as follows:

- Address 10001D (bits set) – Writing a byte to the address sets the selected bits as:

  1 = Set the bit
  0 = No change

- Address 10001F (clear bits) – Writing a byte to the address clears the selected bits as:

  1 = Clear the bit
  0 = No change

5.5.13 Interrupt Control and Counter/Timer Registers

Figure 5–9 shows the register bit settings, hexadecimal addresses, and read/write functions for the interrupt control and counter/timer registers.

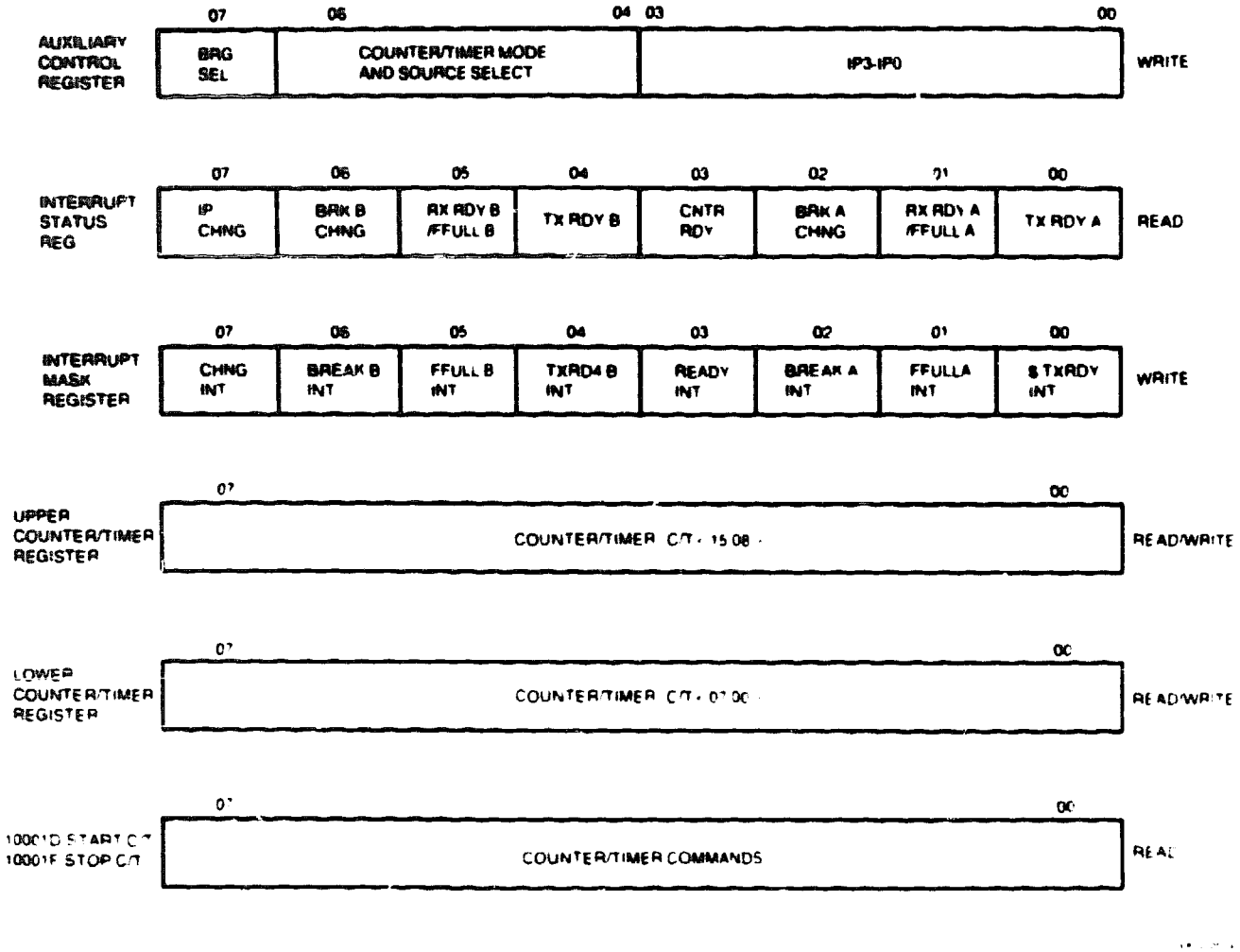


Figure 5–9: DUART Interrupt Control and Counter/Timer Register Settings

5.5.14 Auxiliary Control Register (ACR<06:04>)

Table 5–19 shows the write functions for ACR bits ACR<06:04>, which are used with the counter/timer

Table 5–19: Auxiliary Control Register ACR < 06:04 >

Bits	Name	Description
<07>	-	Used with the clock select registers.
<06:04>	Counter/Timer Mode Source Select and (CTMS SEL)	Sets the counter/timer register to the selected clock source as follows:

Code	Mode	Clock Source
0	Counter	External input (IP2)
1	Counter	TXCA channel A 1X the transmit clock
2	Counter	TXCB channel B 1X the transmit clock
3	Counter	Crystal or external clock divided by 16
4	Timer	External input (IP2)
5	Timer	External input (IP2) divided by 16
6	Timer	Crystal or external clock
7	Timer	Crystal or external clock divided by 16 and used with the input port registers

<03:00> -

5.5.15 Interrupt Status Register (ISR)

Table 5–20 shows the read functions for the ISR status flags. The ISR is a summary register that reflects the states of the interrupt flags.

Interrupt signal output is asserted from the DUART when the corresponding flag bit is set in the IMR.

Table 5–20: Interrupt Status Register (ISR)

Bits	Name	Description
<07>	IP CHNG	Reads as a 1 when any IPCR<07:04> bit is set and the corresponding bit is set in <ACR03:00>. The INTR output is also asserted from the DUART and is unused in the router.
<06>	Channel B BRK B CHNG	Indicates the channel B receiver reached end of a break. Cleared by the reset break change detected in the beginning or at the end of a break command from command register B.
<05>	Receive Ready B or FIFO Full B RX RDYB/FFULL B	Asserts either of the following states, depending on the state of MR1B<06>: MR1B<06> = 0 asserts SRB bit <00> (RX RDY B) MR1B<06> = 1 asserts SRB bit <01> FFULL B
<04>	Transmit Ready B TX RDY B	Asserts the state of SRB bit <02> TX RDY B OP4 - Asserts channel A receive interrupt request.
<03>	Counter Ready	Set by the counter/timer register, depending on the register mode. Counter Mode: Set when the upcount register reaches the count value. Cleared by the stop counter command; stops at the counter. Timer Mode: Set each time the downcount register reaches a zero count. Cleared by the stop counter command; stops the counter. The counter is reloaded from the holding register on each zero count.
<02>	Channel A Change in Break BRK A CHNG	Indicates that the channel A receiver detected the beginning or the end of a break. This is cleared by the reset break change interrupt command from Command Register A.
<01>	Receive Ready A or FIFO Full A RX RDY A/FULL A	Asserts either of the following states, depending on the state of MR1A <06>: MR1A <06> = 0 asserts bit <0>, RXRDY A. MR1A <06> = 1 asserts SRA bit <01>, FFULL A.
<00>	Transmit Ready A	Asserts the state of SRA bit <02>, TX RDY A.

5.5.16 Interrupt Mask Register

Writing a 1 to a bit in the Interrupt Mask Register enables the interrupt signal (INTR) for the corresponding flag in the Interrupt Status Register.

5.5.17 CTUR and CTRL Counter/Timer Registers

The counter/timer has two 16-bit registers, a holding register and a count register. The count register function depends on whether it functions in counter mode or timer mode.

CTUR and CTLR are holding registers that store the upper-byte and the lower-byte values as shown in Figure 5–9. The register addresses are write-only to the holding registers in the counter mode. The register addresses are read only in the timer mode. The minimum load value is 0002 hexadecimal.

Counter Mode – The counter mode is a down counter that sets the interrupt status register bit, <03> CNTR RDY, when the counter reaches its final value of 0000 (hex). Output port OP3 is programmable to assert the ISR state <03>.

The commands affect the counter mode as follows:

- **Stop Counter** – This stops the Count Register and clears ISR <03>.
- **Start Counter** – This loads the Count Register with the holding register contents and starts the count register.

Timer Mode – The Count Register is a binary up-counter that runs continuously, generating a square wave twice the time stored in the holding register.

Each overflow sets interrupt status register bit <03>, CNTR RDY, and reloads the count register from the holding register. OP3 can be programmed to produce a square wave.

The commands affect the timer mode as follows:

- **Stop Timer** – This clears the ISR<03>, but does not stop the count register.
- **Start Timer** – This terminates the timing cycle, loads the count register with the holding register contents, and starts a new timing cycle.

5.5.18 Counter/Timer Start and Stop Commands

Reading either of the following addresses, issues one of these counter/timer commands:

For DUART0:

- Address 10001D – Start counter/timer
- Address 10001F – Stop counter/timer

For DUART1:

- Address 10003D – Start counter/timer
- Address 10003F – Stop counter/timer

For DUART2:

- Address 10006D – Start counter/timer
- Address 10006F – Stop counter/timer

For DUART3:

- Address 10007D – Start counter/timer
- Address 10007F – Stop counter/timer

5.6 Ethernet Interface

Interfacing the DECrouter 200 to the Ethernet is handled with two chips:

- Local Area Network Controller for Ethernet (LANCE)
- Serial Interface Adapter (SIA)

The chip set converts Manchester-encoded serial data to 16-bit parallel data at 10 Mbps.

The following sections summarize the signals and protocol supported by the LANCE and the SIA, including internal register formats and functions.

5.6.1 Serial Interface Adapter (SIA)

Figure 5–10 is a block diagram of the SIA, which shows the interface between the logic signals of the LANCE and the differential signals of the Ethernet transceiver.

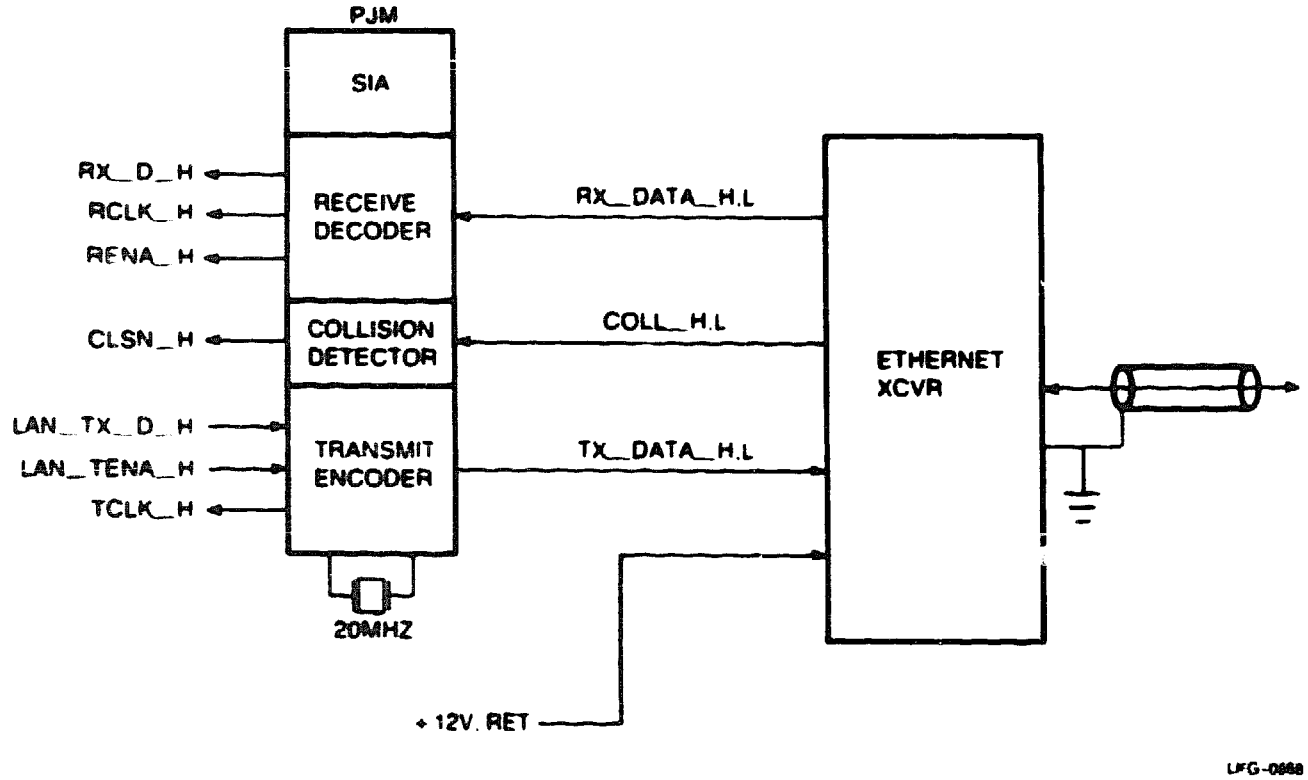


Figure 5–10: SIA Connection to an Ethernet Transceiver

Table 5–21 shows the differential signals passed between the SIA and the Ethernet transceiver on the interface cable.

Table 5–21: SIA and Ethernet Transceiver Interface Signals

Signal Name	Description
Power (+12V, RET)	The power pair supplies regulated 12 Vdc power to the transceiver.
Transmit Data (PJM TX DATA H, L)	The transmit data pair carries a 10-MHz Manchester-encoded differential signal from the SIA transmit encoder to the Ethernet transceiver cable driver.
Receive Data (PJM RX DATA H,L)	The receive data pair carries a 10-MHz Manchester-encoded differential signal from the Ethernet transceiver cable receive to the SIA receiver decoder.
Collision (PJM COLL H,L)	The collision pair carries a 10-MHz collision comparator in the Ethernet transceiver to the SIA collision detector. This occurs on a collision or on completion of a normal transmission, a heartbeat check.

Table 5–22 shows the logic interface signals between the LANCE and the SIA.

Table 5–22: LANCE and SIA Interface Signals

Signal Name	Description
LANCE Receive Signals	
Collision (PJM CLSN H)	Asserted by the SIA to indicate a collision on the Ethernet.
Receive Enable (PJM RENA H)	Asserted by the SIA to indicate a valid signal on the Ethernet. The SIA receiver/Manchester decoder synchronizes and produces the RCLK and RX D signals.
Receive Clock (PJM RCLK H)	A 10-MHz clock produced by the SIA, the synchronizing clock for the LANCE serial data receiver.
Receive Data (PJM RX D H)	A serial data stream produced by the SIA, which is asserted to the LANCE serial data receiver, where it is clocked by RCLK.
LANCE Transmit Signals	
Transmit Clock (PJM TCLK H)	A 10-MHz clock from the SIA, which is the main clock for the LANCE microprocessor and the synchronizing clock for the LANCE serial data transmitter.
Transmit Enable (PJK LAN TX D H)	Asserted by the LANCE to enable the SIA transmitter/driver to the Ethernet transceiver.
Transmit Data (PJK LAN TX D H)	The serial transmit data stream produced by the LANCE, which is asserted to the Manchester data encoder in the SIA, where the data is encoded and sent to the Ethernet transceiver.

A 20-MHz crystal oscillator provides the SIA main timing reference. The timing reference is divided by two, producing a 10-MHz clock as a time base for the LANCE internal state machine and serial data transmitter. The 20-MHz and the 10-MHz clocks both drive the Manchester data encoder to produce the transmissions in the encoded data stream.

5.6.2 Local Area Network Controller for Ethernet (LANCE)

Table 5–23 shows the LANCE data/address bus operating signals. Three-state signals are normally disabled, in high-impedance state.

Table 5–23: LANCE Operating and Data/Address Bus Signals

Signal Name	Description
Data/Address Bus Signals	
Address Bus (BUS<A16:A01>H)	The LANCE uses 16 of the 23 three-state address bus lines to access program RAM as bus master. These are drive-only lines for the LANCE. The LANCE asserts BALE during the address cycle of a DMA transfer. The LANCE forms a bus address by opening the address latch, enabling the three-state outputs, and asserting a memory address on BUS A16 and on the data bus. The LANCE then closes the latch, ending the address cycle and leaving the address asserted on BUS <A16:A01>H.
Data Bus (BUS DAL<15:00>H)	The 16-bit bidirectional three-state data bus is the data path for DMA data transfers by the LANCE, as bus master, or for CPU-initiated register transfers with the LANCE, as a bus slave. As the bus master on a DMA cycle, the LANCE first asserts an address on these lines during the address cycle, and loads it to the address latch. The LANCE then uses the lines for the transfer of data on the data cycle.
Bus Arbitration Signals	
Buffered LANCE Hold (PJK BLANHLD L)	Asserted by the LANCE when requesting access to the data/address bus. BLANHLD initially asserts a bus request to the CPU but is held asserted for the transaction. When negated, LANCE terminates the DMA cycle, releasing the bus.
Bus Grant Acknowledge (PJD BGACK L)	Asserted by the bus arbitrator in response to a bus grant from the CPU.
LANHLD Acknowledge (PJD LHLDA H,L)	Grants bus access to the LANCE for a DMA transfer. LHLDA enables the three-state outputs of the address latch, and asserts LANRDY to the LANCE to indicate that program RAM is ready to perform the transfer.
Asynchronous Bus Control Signals	
LANCE Enable (PJC ENB LANCE L)	Asserted by the CPU to select the LANCE for a register transfer.
Buffered Address Latch Enable (PJK BALE H)	Asserted by the LANCE on the address cycle of a DMA transfer. BALE opens the address latch to receive bits <15:01> of the memory address asserted BUS DAL<15:01>. When negated, BALE closes the latch that holds the address asserted on BUS <A15:A01>.

(continued on next page)

Table 5–23 (cont.): LANCE Operating and Data/Address Bus Signals

Signal Name	Description
Asynchronous Bus Control Signals	
Data Strobe (PJK BDAS L)	Bus Master: Asserted by the LANCE on the data cycle of a CPU transfer. Write data asserted by the LANCE is stable on the high to low transition of BDAS. The LANCE clocks read data on the low-to-high transition of BDAS. Bus Slave: Received from the CPU on the data cycle of a CPU transfer, BDAS selects the LANCE register address port or register data port for the transfer, depending on the state of address bit BUS <01>: BUS <A01> = 0 Register data port BUS <A01> = 0 Register address port
LANCE Ready (PJK LANRDY L)	Bus Master: Asserted by LHLDA on a DMA cycle, LANRDY indicates to the LANCE that program RAM is ready to transfer data. Bus Slave: Asserted by the LANCE on a DMA cycle, LANRDY indicates that the LANCE has asserted read data on the data bus, or has clocked write data from the data bus.
Write (BUS WRT L)	Defines the bus operation as a read or write transfer with respect to the bus master: Write = Asserted low Read = Asserted high As bus master, the LANCE drives this signal on a DMA transfer. As bus slave, LANCE receives the signal on a CPU transfer with a LANCE register.
Upperbyte/Lowerbyte Data Strobe (BUS UDS L/BUS LDS L)	As bus master, the LANCE asserts these signals on DMA transfer to specify the valid byte(s) for a read or a write transfer: Valid Data is on BUS BUS BUS DAL BUS DAL UDS LDS <15:08> <07:00> 0 0 No No 0 1 No Yes 1 0 Yes No 1 1 Yes Yes (0 = negated, 1 = asserted) As bus slave, the LANCE ignores these signals and assumes word transfers. These are wire-OR signals and are driven by the LANCE and the CPU at different times. The signals are buffered in the CPU logic, providing the following outputs:

(continued on next page)

Table 5–23 (cont.): LANCE Operating and Data/Address Bus Signals

Signal Name	Description
Input Buffered Output	BUS WRT L PJA BWRT H,L BUS UDS L PJA BUDS H,L BUS LDS L PJA BLDS H,L
LANCE Interrupt (PJK LANINT L)	Asserted to IPL5 of the priority encoder to request a program interrupt for any six LANCE status flags. Signal is cleared by the interrupt service routine.
Initialize Signals	
Reset	A low signal on the reset input halts the LANCE, clears its logic, and LANCE enters an idle state. - Power-up (PJJ PUP L) from the power-up logic. - LANCE initialize (PJH LINIT L) under program control from DUART 0.

5.6.3 Ethernet Interface Features

The LANCE operates at 10 MHz and performs the data-link-level Ethernet protocol. LANCE features include a storage data FIFO buffer (SILO) that allows back-to-back transfers of up to eight words with program RAM on a single DMA cycle.

After the CPU loads the LANCE control and status registers (CSRs), the LANCE performs a DMA break to program RAM to retrieve the initialize block with the LANCE operating parameters. In operation, the LANCE takes the receive or transmit descriptor rings. The LANCE descriptor rings contain the memory data buffer parameters for transfers between the Ethernet and program RAM through the LANCE.

5.6.3.1 DMA Transfers with Program RAM — The LANCE and the CPU are the only devices that gain access to and that use the data/address bus as a bus master. The LANCE accesses the bus through bus arbitration logic, and performs a direct memory access (DMA) transfer with program RAM.

Bus Arbitration Cycle – The LANCE first arbitrates and accesses the data bus as follows:

1. The LANCE asserts LANCE hold (LAN__HLD__L) to the bus arbitrator, which asserts bus request (BR__L) to the CPU.
2. After releasing the bus, the CPU returns bus grant (BG__L) to the arbitrator.
3. When address strobe (BASTRB__L) is deasserted by the CPU, the arbitrator returns bus grant acknowledge (BGACK__L) to the CPU, which negates bus grant (BG__L).
4. The arbitrator asserts LANCE hold acknowledge (LAN__HLD__ACK__L) to the LANCE and negates bus request (BR__L) to the CPU.
5. With the bus cycle completed, the LANCE negates LAN__DHLD__L and BGACK__L, which releases the bus.

Address Cycle – With bus control, the LANCE initiates selection of a program RAM location as follows:

1. LANCE enables three-state data control outputs — WRT_L, UDS_L, and LDS_L — with the correct states for the transfer.
2. The LANCE enables three-state outputs to the data bus and enables the three-state outputs of its address latch to the address bus.
3. The LANCE asserts a memory address on the data bus and on three-state address bus bit <A16> and opens the address latch by asserting address latch enable (ALE_H).

Data Cycle – The following events transfer data with the selected program RAM location:

1. The LANCE defines the data cycle by asserting buffered data strobe (DAS_L). To write, the LANCE enables its three-state outputs and asserts data on the data bus.
2. The LANCE negates buffered data strobe (DAS_L). To read, LANCE also clocks data from the data bus.
3. The LANCE disables all three-state outputs and puts the signals in high-impedance (high-Z) state.
4. LANCE negates buffered LANCE hold (LAN_HLD_L) to the bus arbitrator, which negates bus grant acknowledge (BGACK_L) and LANCE hold acknowledge (LAN_HLD_ACK_L), terminating the transaction.

5.6.4 Ethernet Operating Modes

The LANCE provides three major operating modes on the Ethernet:

Transmit Mode – In transmit mode, the LANCE initiates a DMA cycle that acquires a transmit descriptor ring from memory. Further DMA cycles acquire data from a transmit buffer in memory and the LANCE shifts the data out to the Ethernet system in serial data frame format.

The LANCE prefaces each data frame with a preamble pattern of binary data, allowing the receive circuits to settle and the Manchester encoders to synchronize. The LANCE calculates a 32-bit cyclic redundancy check (CRC) on the source address and destination address fields, and the type and data fields. The LANCE appends the final CRC value to the end of the data frame. On multiple data-transmissions, the preamble provides an interframe gap of 9.6 microseconds after the CRC.

On data collision, the LANCE increases its defer time according to an exponential *binary backoff algorithm* contained in its internal microprocessor protocol.

Receive Mode – When a carrier is on the Ethernet cable, the preamble allows the SIA receiver circuits to settle. The Manchester decoder synchronizes and produces a separate clock and separate data pulses from the encoded signal carrying the incoming data frame.

The LANCE calculates the following:

- The CRC on the received source address and destination address fields
- The data type and data fields
- The received CRC value

When a calculation produces a 1, an error bit is set in the LANCE, generating an interrupt to the CPU.

Received Address Modes – Each of the following modes compares the received destination address with a pre-selected value, and determines whether the LANCE should accept or should reject the incoming data frame:

1. **Physical address mode** – The data frame is accepted when the destination address matches the router Ethernet node address.
2. **Broadcast address mode** – A destination address of all 1s specifies a broadcast address, and the data frame is always accepted.
3. **Logical address mode** – Accepts all data frames addressed to one type of device or to a set of devices. Addresses are selected using a logical address filter.

5.6.5 Register Address and Data Ports

The CPU selects the LANCE from the address bus. Address bit <A00> must be set to select the lower byte for a word transfer. As a bus slave, the LANCE is only accessed on word boundaries.

BUS <A01> selects either the register address port (RAP) or the register data port (RDP) for a read or a write transfer as follows:

Address 100080 – Selects the RDP

Address 100082 – Selects the RAP

Figure 5–11 shows the register address port and control/status register bit settings. CSR0 has bit settings that allow access to it any time during normal operation. All other CSRs are read/write registers that are only accessible when the STOP bit is set in CSR0.

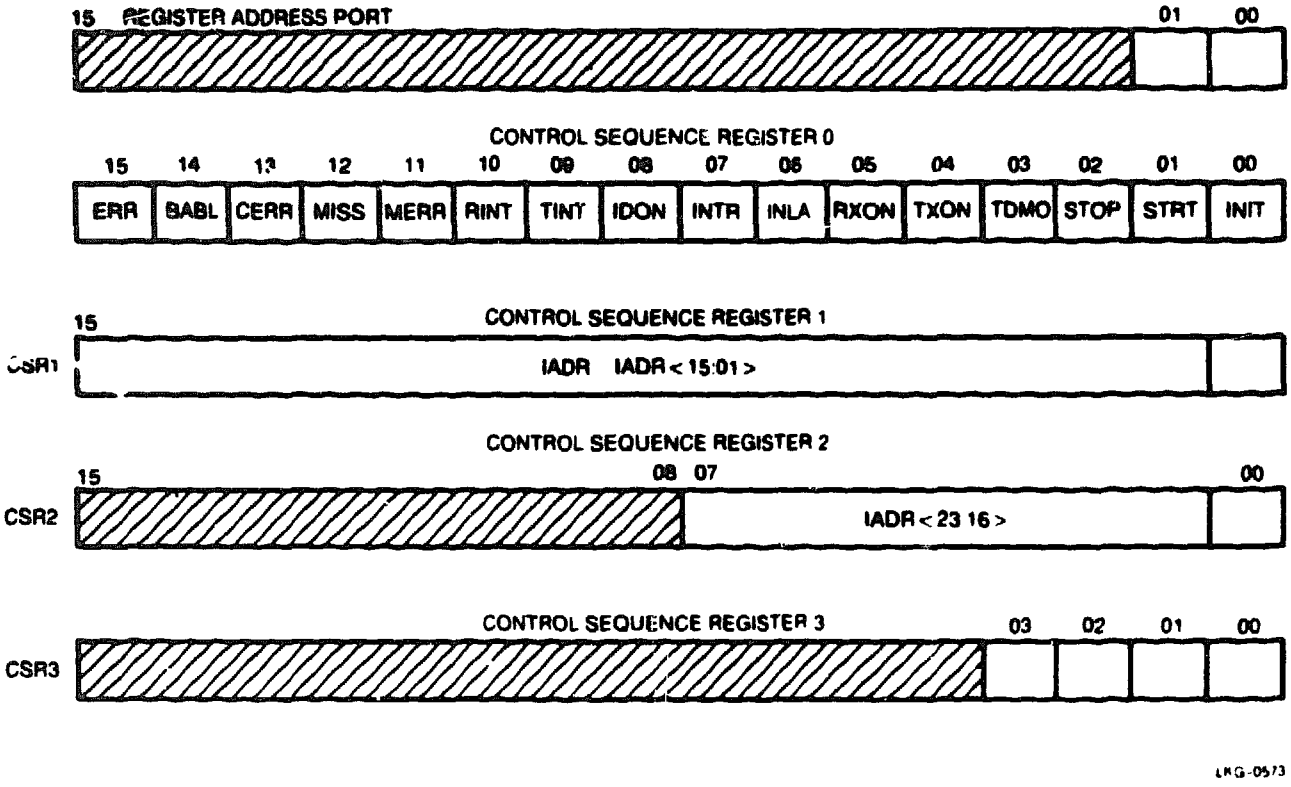


Figure 5–11: LANCE RAP and CSR Bit Settings

5.6.6 Register Address Port (RAP) and Latch

When BUS<A01> is set, the CPU loads the RAP. When BUS<A01> is cleared, it latches and asserts bits RAP<01:00> that select one of four CSRs

The RAP is a read/write register that selects a CSR to access through the RDP. RAP is cleared by STOP or by bus RESET.

5.6.7 Control/Status Register 0 (CSR0)

When address bit <A01> is cleared, the CPU loads the CSR selected by the RAP register through the register data port. All subsequent transfers take place with the selected CSR until the RAP value is changed.

The function of each bit in CSR0, described in the following text, is true when the bit is set. The bits are controlled by the operating program according to these access definitions.

- Read/Write – The bit can be read, can be set, or can be cleared by writing to it.
- Read Only – The bit can only be read.
- Read/Set – The bit is read-only but can be set by writing a 1 to it.

- Read/Clear – The bit is read-only but can be cleared by writing a 1 to it.
- Write Only – This is a write-only bit and a LANCE function is done by writing a 1 to it.

All register bits are cleared by setting the STOP bit or by asserting the reset input except for bit <02>, which is the STOP bit in CSR0. CSR1, CSR2, and CSR3 can only be read to when the STOP bit in CSR0 is set. These CSRs must also be initialized after a reset or stop operation.

Table 5–24 identifies, gives the mnemonic and bit-types and describes the function of all the bits in CSR0.

Table 5–24: Control and Status Register 0 (CSR0)

Bit	Name	Description	Bit Type
<15>	Error (ERR)	Asserts an OR summary of error bits <14:11> Read as 0 when bits are clear	Read-Only
<14>	BABL	Indicates a transmit error. The transmitter is enabled longer than necessary to send the 1529-byte packet maximum	Read/Clear
<13>	CERR	Indicates a collision error. The collision test signal, the heartbeat, was not acquired within 2 microseconds of a normal data transmission	Read/Clear
<12>	MISS	Indicates a missed data packet. The receiver lost a data packet due to lack of receive buffer acquisition. SILO is overflowed	Read/Clear
<11>	MERR	Indicates a memory error. LANCE did not receive LANRDY within 25.6 microseconds of address assertion on the data bus	Read/Clear
<10>	RINT	Indicates a receiver interrupt. The bit is set after the LANCE updates an entry in the receive descriptor ring	Read/Clear
<09>	TINT	Indicates a transmit interrupt. The bit is set after the LANCE updates an entry in the transmit descriptor ring	
<08>	IDON	Indicates initialize done. The LANCE has read the initialization block and has stored the parameters	Read/Clear
<07>	INTR	Identifies an interrupt condition for BABL, MISS, MERR, RINT, TINT, or IDON	Read-Only
<06>	INEA	Enables an interrupt when <07> is set	Read/Write
<05>	RXON	Indicates that the receiver is enabled. The bit is set when STRT is active and DRX is 0 in the mode register	Read-Only

(continued on next page)

Table 5–24 (cont.): Control and Status Register 0 (CSR0)

Bit	Name	Description	Bit Type
<04>	TXON	Indicates the transmitter is enabled. The bit is set when STRT is enabled, DTX is 0 in the mode register in the initialization block, and the INIT bit is set in CSR0.	Read-Only
<03>	TDMD	Indicates transmit demand. When set, the LANCE acquires the transmit descriptor ring without waiting for a polltime interval to elapse.	Read-Only
<02>	STOP	STOP bit for the LANCE. When set, all external activity is disabled and internal logic is cleared.	Read/Set
<01>	STRT	Start bit. Start enables the LANCE to send and receive data packets, to perform direct memory access, and to manage the buffer	Read/Set
<00>	INIT	Initialize bit. Setting this bit enables LANCE initialization, and access to the LANCE initialize block	Read/Set

5.6.8 Control and Status Register 1 (CSR1)

CSR1<15:01> stores the lower bits of the first initialize block address. Table 5–25 identifies, gives the mnemonic and describes the functions of all bits in CSR1.

Table 5–25: Control and Status Register 1 (CSR1)

Bits	Name/Description
<15:01>	IADR. The low order 16 bits of the address of the first word, lowest address, in the initialize block
<00>	MBZ. This bit must be zero and is not used

5.6.9 Control and Status Register 2 (CSR2)

CSR2<07:00> stores the upper bits of the first initialize block address. Table 5–26 identifies, gives the mnemonic and describes the functions of all bits in CSR2.

Table 5–26: Control and Status Register 2 (CSR2)

Bits	Name/Description
<15:08>	Reserved
<07:00>	IADR. The high order 8 bits of the address of the first word, lowest address, in the initialize block.

5.6.10 Control and Status Register 3 (CSR3)

CSR3 controls the byte swap functions and pin definitions of the bus master interface. Table 5–27 identifies, gives the mnemonic and describes the functions of all bits in CSR3.

Table 5–27: Control and Status Register 3 (CSR3)

Bits	Name	Description
<15:03>	-	Reserved.
<02>	BWSP	Byte Swap allows the LANCE to start on an odd byte address. Data bits <15:01> are swapped with <07:00> on a DMA transfer.
<01>	ACON	Address Line Enable defines the asserted state of the LANCE from the STOP bit as: ACON = 0 ALE is asserted high ACON = 1 ALE is asserted low
<00>	BCON	Byte Control redefines the byte mask and holds I/O pins as: Pin 17 Pin 16 Pin 15 BCON = 0 LANHLD BUS UDS BUS LDS BCON = 1 BUSACK BYTE BUSRQ

5.6.11 Buffer Management Protocol

The router operating software maintains data buffer areas in memory for the temporary storage of Ethernet data from the LANCE. The software also contains tables in memory that LANCE acquires under DMA. LANCE uses these tables to access the data buffers under DMA transfers.

The LANCE buffer management protocol uses several table structures to handle the data table and the memory buffers, including:

- **Data buffers** – The receive and transmit data buffers reside in contiguous memory space and can start on arbitrary byte boundaries.
- **Initialize block** – The initialize block is a table in memory that stores the LANCE operating mode, the addressing parameters, and the pointers to the base addresses for the receive and transmit descriptor rings. The initialize block starts on a two-byte boundary.
- **Descriptor rings** – The receive and transmit descriptor rings are tables in memory that LANCE uses to access the receive and transmit data buffers in memory. Each descriptor ring has at least one entry and starts on a four-word boundary.

  The descriptor rings provide an information path between the LANCE and the operating program for filling and emptying a data buffer in memory.
- **Message descriptors** – Each descriptor ring entry has four words called message descriptors. Message descriptors supply the following information to access one buffer:
  - Point to the base address of the buffer
  - Specify the length of the buffer

- Specify method of data frame processing
- Specify LANCE action on error or on change in status

Multiple entries are required when a transfer exceeds the size of the buffer in memory. Multiple data frame transmissions are required for transfers that exceed the maximum data frame length.

Router operations begin when the operating program sets the LANCE initialize bit (INIT CSR0<00>). The LANCE acquires the initialize block from memory, starting at the address specified in CSR1 and CSR2, and stores the information in its internal parameter registers. This enables the LANCE to start transfer operations when the operating program assembles the receive and the transmit descriptor rings in memory.

5.6.12 Initialize Block

The initialize block has 12 contiguous words in memory, starting on a two-byte boundary, as shown in Figure 5–12.

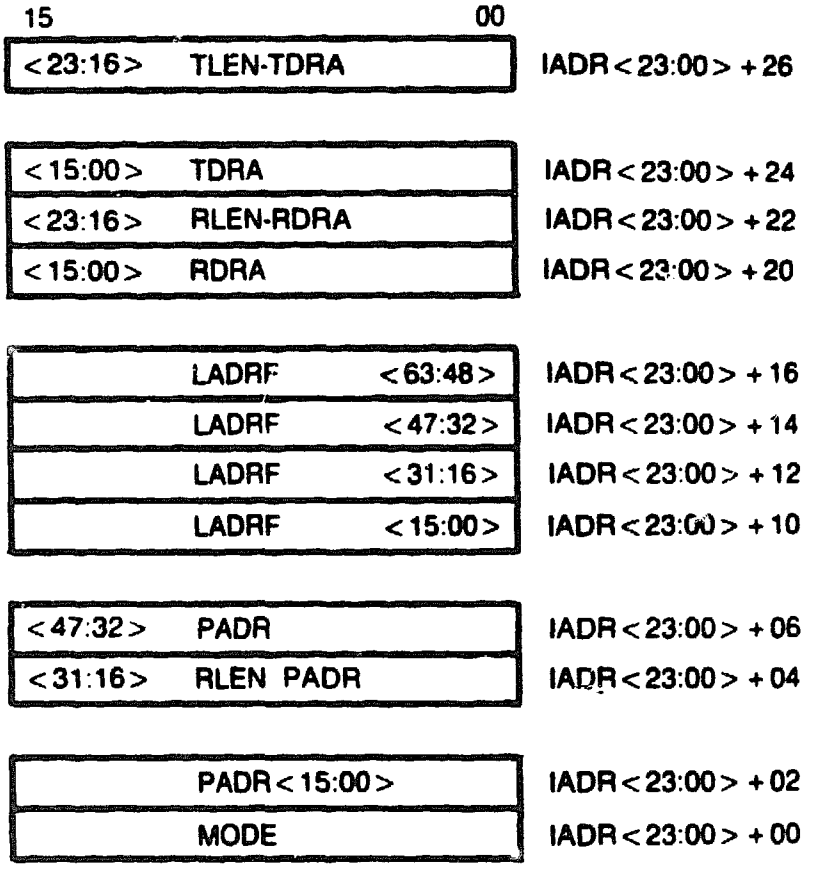


Figure 5–12: LANCE Initialize Block Format

The router software assembles the initialize block before initializing the LANCE. When initialized, the LANCE performs a DMA cycle that acquires the following operating parameters from the initialize block and stores the parameters in registers.

- 1 word – Mode (MODE)
- 3 words – Physical Address (PADR)
- 4 words – Logical Address Filter (LADRF)
- 2 words – Receive Descriptor Ring Address (RDRA)
- 2 words – Transmit Descriptor Ring Address (TDRA)

Station Addressing – The LANCE uses the Physical Address (PADR) Register and Logical Address Filter (LADRF) register to determine whether to accept or to reject incoming data frames.

- Physical address – When the first bit of an incoming address is 0, the incoming address is physical. If the address matches the contents of the PADR register, the data frame is accepted.
- Broadcast address – The first bit must be 1 for the broadcast address decoder or the logical address filter to be enabled. A destination address with all 1s is the broadcast and the data frame is always accepted.
- Logical address – For any other address where the first bit is 1, data frames are accepted according to the LADRF register. The operating software handles final filtering.

  When all 48 bits of an incoming address pass through the CRC circuitry, the six high-order bits of the 32-bit CRC are clocked to a register that selects one of 64 bit positions in the LADRF register. When the selected filter bit is a 1, the data frame is accepted.

  When the LADRF is all 0s, all incoming logical addresses are rejected except for the broadcast address.

Minimum Data Frame Length – Incoming data messages must be at least 64 bytes long to be valid. Messages shorter than 64 bytes are not updated in the LANCE receive message descriptor pointing to the buffer. The contents are saved for the next data frame.

5.6.13 Mode Register (MODE)

MODE<15:00> bits are cleared during normal operation. These bits allow the LANCE operating parameters to change for maintenance functions, as shown in Table 5–28.

Table 5–28: Mode Register Bit Settings

Bits	Name	Description
<15>	Promiscuous PROM	When this bit is set, the LANCE accepts all incoming data frames with any destination address.
<14:07>	–	Reserved.
<06>	Internal Loopback INTL	This is valid only when bit <02> LOOP is set; INTL selects the LANCE loopback path: INTL = 0 External loopback INTL = 1 Internal loopback With INTL set, the data frame size is limited by the 48-byte SILO. The data is 32 bytes long with CRC disabled, and 36 bytes long with CRC enabled.
<05>	Disable Retry DRTY	When set, the LANCE tries only one transmission. A collision on a first try, sets the retry error bit, RTRY <10>, in transmit message descriptor 3, TMD3.
<04>	Force Collision FCOL	This is valid only in internal loopback mode, and is set for testing the collision logic. The LANCE makes 16 transmission attempts before setting the retry error bit, RTRY <10> in TDM3.
<03>	Disable Transmit CRC DTCR	When clear, the transmitter generates a CRC and appends the CRC to the data frame. On a loopback test with DTCR clear, the receiver cannot check the CRC. The CRC is written to memory and checked by software. When set, the CRC logic is used by the receiver. On a loopback test with DTCR set, the software must generate and append the CRC. The receiver CRC logic checks and reports any receive errors.
<02>	Loopback LOOP	Sets the LANCE operation to full-duplex mode for testing. Due to the 64-byte minimum on Ethernet data frames, the minimum data frame filter is disabled. The LANCE holds until the full message is in SILO before data transmission. Incoming data follows outgoing data and is passed to memory when the transmission is complete.
<01>	Disable transmitter DTX	When set, this disables LANCE access to the transmit descriptor ring. The CSR0 bit <10> TXON is disabled and transmits are not attempted.
<00>	Disable Receiver DRX	When set, this disables LANCE access to the receive descriptor ring. The CSR0 bit <05> RXON is disabled and all incoming data frames are rejected.

5.6.14 Physical Address Register (PADR)

The PADR<47:00> bits store the router Ethernet node address that was generated by the PA PROM. PADR<00> must be 0.

5.6.14.1 Logical Address Filter Register (LADRF) — LADRF<63:00> bits form a 64-bit mask used to accept incoming logical addresses.

5.6.14.2 Receive Descriptor Ring Address Register (RDRA) — The RDRA<31:00> bits are described in Table 5–29.

Table 5–29: Receive Descriptor Ring Address (RDRA) Bit Settings

Bits	Name	Description
<15:13>	Receive Ring Length (RLEN <2:0>)	The three high-order bits specify the number of receive descriptor ring entries as a power of two:

RLEN Bits <2:0>	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

<12:08>	-	Reserved.
<07:00>	Receive Ring Address (RDRA<15:03>)	Specify the receive descriptor ring base address.
<02:00>	Must be Zeros (MBZ)	Receive descriptor rings are aligned on four-word boundaries

5.6.14.3 Transmit Descriptor Ring Address Register (TDRA) — The TDRA<31:00> bits have the functions described in Table 5–30.

Table 5–30: Transmit Descriptor Ring Address (TDRA) Bit Settings

Bits	Name	Description
<15:13>	Transmit Ring Length (TLN<2:0>)	The three high-order bits specify the number of transmit descriptor ring entries as a power of two:

TLEN Bits <2:0>	Number of Entries
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128

<12:08>	–	Reserved.
<07:00>	Transmit Ring Address (RDRA<15:03>)	Specify the transmit descriptor ring base address.
<02:00>	Must be Zeros (MBZ)	Transmit descriptor rings are aligned on four-word boundaries.

5.6.15 Receive Descriptor Ring

A receive descriptor ring has at least one entry in memory, as shown in Figure 5–13. Each entry points to a receive data buffer and defines how data is used.

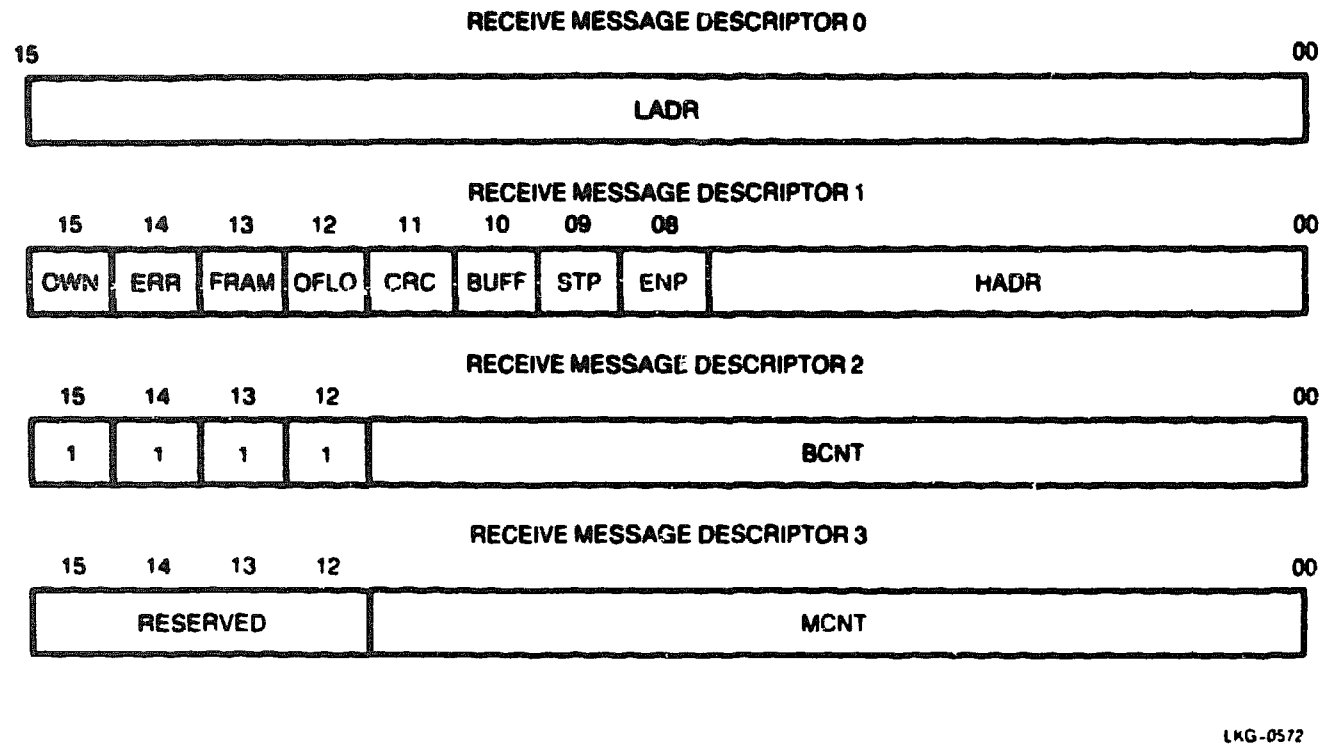


Figure 5–13: LANCE Receive Descriptor Ring Entry

Each entry has four words called receive message descriptors (RMDs) and is accessed on a four-word boundary. The following four tables show the bit settings in each descriptor.

Table 5–31: Receive Message Descriptor 0 Bit Functions

Bits	Name	Description
<15:00>	Low Address LADR<15:00>	These bits are the low-order base address bits for the receive data buffer selected by the descriptor ring entry. LADR is written by the host and is unchanged by the LANCE

Table 5–32: Receive Message Descriptor 1 Bit Functions

Bits	Name	Description
<15>	OWN	This bit indicates ownership of the entry: 0 = Owned by HOST, router software 1 = Owned by LANCE The following bits are set by LANCE and cleared by the HOST:
<14>	ERR	An OR summary of error status bits <13:10>.
<13>	FRAM	This indicates that the received data frame has a noninteger multiple of eight bits and a CRC error.
<12>	OFLO	Indicates that all or part of a received data packet was lost when LANCE could not access the buffer and store the data before the SILO overflowed.
<11>	CRC	Indicates that a CRC error was detected on a data frame.
<10>	BUFF	Indicates the LANCE has used all allocated buffers or did not acquire the next entry status before a SILO overflow.
<09>	STP	This indicates that an entry has the pointer to the first data buffer for the packet. This is used for chaining buffers.
<08>	ENP	This indicates the last buffer used by LANCE for the packet. With STP and ENP both set in one entry, the data packet fits into one buffer and no chaining is in effect.
<07:00>	HADR	These are the 8 high-order address bits of the buffer pointed to by the descriptor. This is written by the HOST.

Table 5–33: Receive Message Descriptor 2 Bit Functions

Bits	Name	Description
<15:12>	MBO	This field is written by the HOST and is unaffected by LANCE.
<11:00>	BCNT	BCNT is the length of the buffer pointed to by the descriptor expressed as a two's complement number. Field is written by the HOST and is unaffected by LANCE.

Table 5–34: Receive Message Descriptor 3 Bit Functions

Bits	Name	Description
<15:12>	-	Reserved and read as zeros.
<11:00>	MCNT	MCNT is the length, in bytes, of the received message. This is written by the LANCE and is cleared by the HOST.

5.6.16 Transmit Descriptor Ring

A transmit descriptor ring has a minimum of one entry in memory, as shown in Figure 5–14.

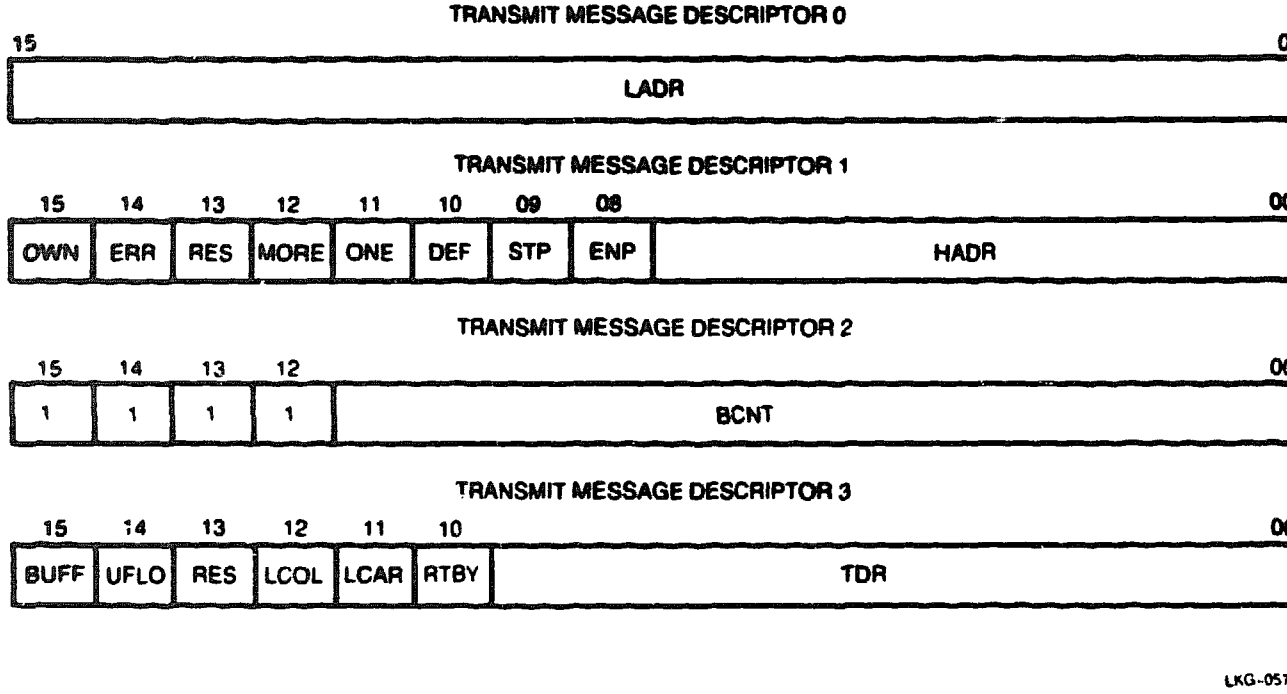


Figure 5–14: LANCE Transmit Descriptor Ring

The following four tables describe the bit settings in the four transmit descriptor rings.

Table 5–35: Transmit Message Descriptor 0 Bit Functions

Bits	Name	Description
<15:00>	LADR<15:00>	These bits are the low-order base address bits for the transmit data buffer selected by the descriptor ring entry. LADR is written by the host and is unchanged by the LANCE

Table 5–36: Transmit Message Descriptor 1 Bit Functions

Bits	Name	Description
<15>	OWN	This bit indicates ownership of the entry: 0 = Owned by HOST, router software 1 = Owned by LANCE The HOST sets the OWN bit after filling the transmit data buffer. The LANCE clears the buffer after transmitting the packet. The following bits are set by LANCE and cleared by the HOST.
<14>	ERR	An OR summary of LCOL, LCAR, UFLO, or RTRY. ERR is set by LANCE and cleared by the HOST.
<13>	-	Reserved, LANCE writes this bit as a zero.
<12>	MORE	This indicates that more than one retry was needed to send a packet.
<11>	ONE	This indicates that a transmission was completed on one try.
<10>	DEF	This indicates that LANCE has deferred a transmission due to Ethernet traffic.
<09>	STP	This indicates that an entry has the pointer to the first data buffer for the packet. This is used for chaining buffers.
<08>	ENP	This indicates the last buffer used by LANCE for the packet. With STP and ENP both set in one entry, the data packet fits into one buffer and no chaining is in effect.
<07:00>	HADR	These are the 8 high-order address bits of the buffer pointed to by the descriptor. This is written by the HOST.

Table 5–37: Transmit Message Descriptor 2 Bit Functions

Bits	Name	Description
<15:12>	MB0	This field is written by the HOST and is unaffected by LANCE
<11:00>	BCNT	BCNT is the length of the buffer pointed to by the descriptor expressed as a two's complement number. Field is written by the HOST and is unaffected by LANCE.

Table 5–38: Transmit Message Descriptor 3 Bit Functions

Bits	Name	Description
<15>	BUFF	This bit is set by LANCE during transmission when the ENF TMD2 bit<08> is not set for the buffer and LANCE does not use the next buffer
<14>	UFLO	This indicates that data was lost due to lack of data from memory. This indicates that SILO has emptied before the end of a packet was received
<13>	Reserved	LANCE writes this as zero.
<12>	Late Collision	This indicates a collision after the allowed slot time.
<11>	LCAR	This indicates that the carrier input negated during LANCE initialization. The chip will not retry on loss of carrier.
<10>	RTRY	This indicates that LANCE has tried and failed 16 times to transmit a packet. When RTRY is a 1, in the mode register, retry sets after one failure.
<09:00>	TDR	This indicates the state of an internal LANCE counter from the start of a packet transmission to a collision detection. The state of the counter is useful for determining the approximate distance to a coaxial cable fault.


СНАР 6

6
Hardware Description

6.1 General

The DECrouter 200 system is a stand-alone communications device. The router can be wall-mounted or table-top mounted. Table-top mounting requires a minimum of 6 inches clearance on all sides for ventilation, and ample clearance in the back for cable access and servicing.

For detailed instructions on installation and site preparation, see the *DECrouter 200 Hardware Installation/Owner's Guide*.

6.2 Port Devices Supported By DECrouter 200

This section lists port devices supported by the DECrouter 200 system. For the latest listing of supported devices, see the DECrouter 200 Software Product Description that applies to your operating system.

Devices supported by the DECrouter 200 system include

- Personal Computers — Digital and non-Digital personal computers running DECnet Phase IV, including:
  - Professional 300 series
  - Rainbow 100 series
- Non-Digital personal computers, including:
  - IBM PC
  - IBM PC/XT
  - IBM Personal Computer AT

- **Modems** — Full-duplex modem control and all asynchronous, full-duplex Digital modems in both dial-in and dial-out modes, including:
  - DF02 (300 bps)
  - DF03 (300/1200 bps)
  - DF112 (300/1200 bps)
  - DF124 (1200/2400 bps)
  - DF224 (300/600/1200/2400 bps)

  Non-Digital modems supported include modems compatible with BELL 103J and BELL 212A, and modems that conform to CCITT V.21, V.21 bis, V.22, and V.22 bis.

6.2.1 Ordering Information

Table 6–1 lists order codes for DECrouter 200 related hardware products. See your Digital sales representative to purchase equipment.

For a listing of software options, see the DECrouter 200 Software Product Description that applies to your operating system.

Table 6–1: DECrouter 200 Hardware Units

Description	Order Code
120 Vac (includes DSRVC-KA country kit)	DSRVC-AA
240 Vac	DSRVC-AB

6.2.2 DECrouter 200 Country Kits

Each of the following kits includes a power cord, the documentation, an Ethernet loopback connector, and a rack mount kit. Table 6–2 shows the order codes for DECrouter 200 country kits.

Table 6–2: DECrouter 200 Country Kits

Country	Order Code
Australia	DSRVC-KZ
Belgium	DSRCV-LA
Canada (English and French)	DSRVC-KA
Denmark	DSRVC-KD
Finland	DSRVC-LA
France	DSRVC-LA
Germany	DSRVC-KG

(continued on next page)

Table 6–2 (cont.): DECrouter 200 Country Kits

Country	Order Code
Holland	DSRVC-LA
Italy	DSRVC-KI
India	DSRVC-LJ
Israel	DSRVC-KT
Japan	DSRVC-KJ
New Zealand	DSRVC-KZ
Norway	DSRVC-LA
Spain	DSRVC-LA
Sweden	DSRVC-LA
Switzerland (French and German)	DSRVC-LB
United Kingdom	DSRVC-KE
United States	DSRVC-KA

6.3 DECrouter 200 Accessories

Table 6–3 gives the router accessories by order numbers.

Table 6–3: DECrouter 200 Accessories

Description	Order code
Ethernet turnaround connector — For testing transceiver and transceiver cable	H4080
Ethernet loopback connector — For loopback testing the DECrouter 200 Ethernet port and transceiver cable	12-22196-01
Port loopback connector — For loopback testing the DECrouter 200 device ports	12-15336-08
Etherjack kit — For covering and securing transceiver cable connections	DEXJK
Wall/partition mounting bracket kit — For mounting the DECrouter 200 to walls or office partitions	H039
Rack mount kit — For mounting the DECrouter 200 in standard rack cabinets	H041-AA

6.4 Transceiver Cables

The BNE3x-xx transceiver cable is available in FEP versions, for use in return air conduits, and in PVC versions, for use in nonenvironmental airspaces. The large diameter of this cable results in a lower signal loss per length of cable than the smaller diameter office transceiver cable. Two styles of connectors are available: a straight connector and a right-angle connector.

The following cables are available:

- BNE3A-xx PVC, straight-connector transceiver cable
- BNE3B-xx PVC, right-angle connector transceiver cable
- BNE3C-xx FEP, straight-connector transceiver cable
- BNE3D-xx FEP, right-angle connector transceiver cable
- BNE3H-xx PVC, straight-connector, 802.3-compliant transceiver cable
- BNE3K-xx PVC, right-angle connector, 802.3-compliant transceiver cable
- BNE3L-xx FEP, straight-connector, 802.3-compliant transceiver cable
- BNE3M-xx FEP, right-angle connector, 802.3-compliant transceiver cable

All these cables are available in lengths of 5 meters (16.4 f[illegible]), 10 meters (32.8 feet), 20 meters (65.6 feet), and 40 meters (131.2 feet).

The BNE4x-xx office transceiver cable is available in PVC versions for use in nonenvironmental airspaces. The smaller diameter of this cable makes it suitable for use in office environments. The smaller diameter of this cable results in a cable signal loss that is four times greater than that of BNE3x-xx transceiver cables. Two styles of connectors are available: a straight connector and a right-angle connector.

The following cables are available:

- BNE4A-xx PVC, straight-connector transceiver cable
- BNE4B-xx PVC, right-angle connector transceiver cable
- BNE4C-xx PVC, straight-connector, 802.3-compliant transceiver cable
- BNE4D-xx PVC, right-angle connector, 802.3-compliant transceiver cable

The preceding cables are available in lengths of 2 meters (6.6 feet) and 5 meters (16.4 feet).

6.5 Device Cables

The following device cables are available:

Null modem cable, round, 10-wire, fully shielded, EIA RS-232-C/CCITT V.28, female-to-female molded connectors:

Length	Order Code
02 ft (0.6 m)	BC17D-02
10 ft (3.0 m)	BC17D-10
25 ft (7.6 m)	BC17D-25
50 ft (15.2 m)	BC17D-50
100 ft (30.5 m)	BC17D-A0

Null modem cable, round, 6-wire, fully shielded, EIA RS-232-C/CCITT V.28, female-to-female molded connectors:

Length	Order Code
10 ft (3.0 m)	BC22D-10
25 ft (7.6 m)	BC22D-25
35 ft (10.7 m)	BC22D-35
50 ft (15.2 m)	BC22D-50
75 ft (22.9 m)	BC22D-75
100 ft (30.5 m)	BC22D-A0
150 ft (45.7 m)	BC22D-A5
200 ft (61.0 m)	BC22D-B0
250 ft (76.2 m)	BC22D-B5

Modem cable, round, 16-wire, fully shielded, EIA RS-232-C/CCITT V.28, male-to-female molded connectors:

Length	Order Code
10 ft (3.0 m)	BC22E-10
25 ft (7.6 m)	BC22E-25
35 ft (10.7 m)	BC22E-35
50 ft (15.2 m)	BC22E-50
75 ft (22.9 m)	BC22E-75
100 ft (30.5 m)	BC22E-A0
150 ft (45.7 m)	BC22E-A5
200 ft (61.0 m)	BC22E-B0
250 ft (76.2 m)	BC22E-B5

Full modem cable, round, 25-wire, fully shielded, EIA RS-232-C/CCITT V.28, male-to-female molded connectors:

Length	Order Code
10 ft (3.0 m)	BC22F-10
25 ft (7.6 m)	BC22F-25
35 ft (10.7 m)	BC22F-35
50 ft (15.2 m)	BC22F-50
75 ft (22.9 m)	BC22F-75
100 ft (30.5 m)	BC22F-A0
150 ft (45.7 m)	BC22F-A5
200 ft (61.0 m)	BC22F-B0
250 ft (76.2 m)	BC22F-B5

6.6 DECrouter 200 Specifications

This section lists the DECrouter 200 specifications.

6.6.1 Power

Table 6–4 shows the power requirements for the DECrouter 200 system.

Table 6–4: DECrouter 200 Power Ratings

Requirements	DSRVC-AA	DSRVC-AB
Factory-set nominal voltage	100 Vac to 120 Vac 3-wire, single phase	220 Vac to 240 Vac 1N + PE
Frequency	47 Hz to 63 Hz	47 Hz to 63 Hz
Line current	1.0 A	0.5 A
Power	75 watts	75 watts

6.6.2 Environment

The following three sections provide environmental specifications for the router.

6.6.2.1 Temperature — Operating: 5° C to 50° C (41° F to 122° F)

Nonoperating: –40° C to 66° C (–40° F to 151° F)

Maximum temperature change per hour: 20° C (36° F)

Rapid temperature changes can affect operation. A router should not be mounted near heating or cooling devices, large windows, or doors that open to the outside.

6.6.2.2 Altitude — Operating: 2.4 km (8000 ft)

Nonoperating: 9.1 km (30,000 ft)

If the router is operating above 2.4 kilometers, the operating temperature must be decreased by 1.8°Centigrade/1000 meters (1° Fahrenheit/1000 feet).

6.6.2.3 Relative Humidity — Operating: 10% to 95% (noncondensing)

Nonoperating: 95% maximum

Low humidity can cause static electricity that can affect router operation. A humidifier is suitable to correct the dew point of an environment.

6.6.2.4 Physical Dimensions of the DECrouter 200 System — Following are the dimensions of the router.

Width: 49.3 cm (19.4 in.)

Height: 11.7 cm (4.6 in.)

Depth: 31.2 cm (12.3 in.)

Weight: 5.9 kg (13.0 lb)

6.6.3 Space Requirements

Allow for 15 centimeters (6 inches) of airspace around the router air vents, and place the router at least 45 centimeters (18 inches) above the floor. This allows adequate ventilation for cooling fans and reduces exposure to excess dust from foot traffic.

Figure 6–1 shows the DECrouter 200 system dimensions.

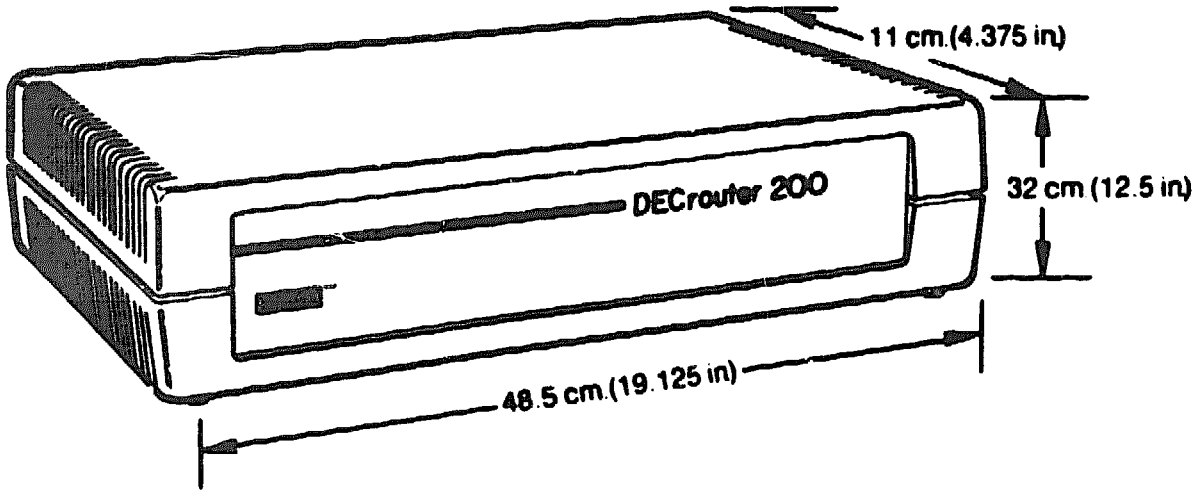


Figure 6–1: Dimensions of the DECrouter 200 System


INDEX

Index

S

T

U

W